

SEP EBPP GATEWAY

An Integrated National Electronic Bill Presentment and Payment (EBPP) Gateway in Syria

Biller Integration Guide Web service

Version: 1.3
Date: 04/03/2020

RIGHTS NOTICE

All rights reserved. These materials are confidential and proprietary to Syrian Electronic Payment Co. (SEP). No part of these materials should be reproduced, published in any form by any means, electronic or mechanical including photocopy or any information storage or retrieval system nor should the materials be disclosed to third parties without the express written authorization of SEP.

Table of Contents

1	System architecture.....	4
1.1	SEP Platform Description	4
1.2	SEP Platform Objectives	5
1.3	SEP Platform Stakeholders	6
2	Bill presentment & payment workflow	6
2.1	Bill Upload	6
2.1.1	Bill Upload Push Mode	7
2.1.2	Bill Upload Pull Mode.....	8
2.2	Bill Presentment	10
2.3	Bill Payment	12
3	Billers Integration Process.....	15
3.1	Integration Approach	15
3.1.1	Integration Technology/Protocol	15
3.1.1.1	ISO 8583 Protocol	15
3.1.1.2	Web Service	17
3.1.2	Integration Workflow	19
3.1.3	Integration Security	22
3.1.3.1	Getting Publics Certificates.....	23
3.1.3.2	Invoke Web Services.....	28
3.1.4	Enumerated Types	32
3.1.4.1	Biller Categories.....	33
3.1.4.2	Service Types	33
3.1.4.3	Pre-Paid Denomination	33
3.1.4.4	Bill Types	33
3.1.4.5	Payment Statuses.....	34
3.1.4.6	Bill Statuses.....	34
3.1.4.7	Access Channels	35
3.1.4.8	Payment Methods.....	36
3.1.4.9	Payment Types	36
3.1.4.10	Billing Number Statuses	36

3.1.4.11	Official ID Types	36
3.1.4.12	Customer Profile Nationalities	37
3.1.4.13	Payment Currencies	37
4	SOAP REQUEST and RESPONSE STRUCTURE	37
4.1	WS Request Structure	37
4.2	WS Response Structure	38
5	Digital signature management	40
5.1	Generation of signature for request message	40
5.2	Verify the signature of the request message	42
5.3	Generation of signature for response message	43
5.4	Verify the signature of the response message	45
6	Biller Interface Specification	46
6.1	Biller Interface Based On Web Service	46
6.1.1	Customer Authentication	46
6.1.2	Post-Paid Electronic Bill Presentment & Payment	47
6.1.2.1	Bill Upload	47
6.1.2.1.1	Bill Upload Push Mode	47
6.1.2.1.2	Bill Upload Pull Mode	50
6.1.2.2	Bill Presentment	53
6.1.2.3	Bill Payment Request	57
6.1.2.4	Payment Notification	59
6.1.3	Pre-Paid Validation	61
6.1.3.1	Pre-paid Inquiry	61
6.1.3.2	Pre-paid Payment	62
6.1.3.3	Pre-Paid Payment Notification	64

1 System architecture

1.1 SEP Platform Description

The Syrian Electronic Payment Company (SEP) was established as a joint-stock company in 2012. The primary function of SEP is to be a central EBPP entity connecting all banks and billers in the country to an integrated infrastructure.

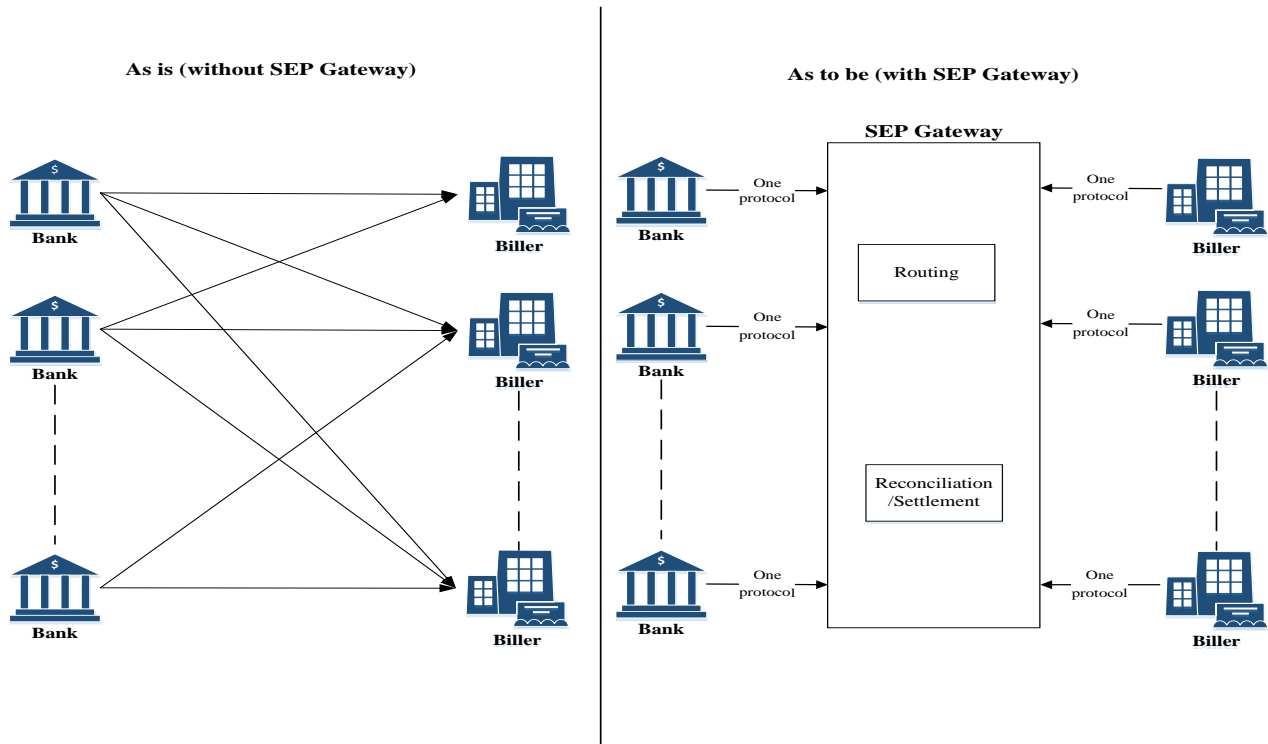
SEP platform is designed to be the single access point for all banks and billers in order to provide the bill presentment and payment service to the end customer. This configuration is considered as a remedy of the current configuration based on a complicated web of connections that each bank or biller must build in order to provide the same service that will be provided in SEP configuration.

The EBPP gateway is expected to provide a mechanism to handle all payment transactions of end customers for periodic bills (water, electricity, fixed-line phone, etc.) and one-off initiated payments (traffic fines, taxes, air tickets, etc.) through all channels of Syria's banking system.

This platform should consist of the following module/functions:

- Interface with banks that will allow both bank and SEP to exchange any message used for bill presentment and payment.
- Interface with billers that will allow both biller and SEP to exchange any message used for bill presentment and payment.

1.2 SEP Platform Objectives



In order to provide electronic presentment and payment service to the end customer so that he can query and pay his bills through bank's channels (ATM, POS, Internet, etc...), currently billers and banks need to build one-to-one linkage that consists of the following tasks:

- Implementation of an interface between biller and bank for bill presentment and payment that should respect a given protocol.
- Implementation of reconciliation process that will allow to match the payments between the two parties.
- Definition of the dispute regulation that should be respected in case of discrepancy/inconsistency or litigious situations.

Each linkage a given bank and biller need to build generates an additional cost in terms of operation and project management which make both banks and billers more reluctant to provide this service at a large scale, and the expansion of the electronic bill presentment and payment is very limited.

SEP Gateway as a platform will be implemented to eradicate the constraints mentioned above and to allow a wide expansion of electronic bill presentment and payment since both banks and billers will benefit of the following advantages:

- One interface/protocol will be implemented by the billers/banks to open the service many-to-many through SEP platform.
- Reconciliation and Settlement will be performed by SEP on behalf of banks and billers.
- Arbitration in case of any discrepancy and litigious situation will be conducted by SEP.

1.3 SEP Platform Stakeholders

EBPP gateway is expected to bring distinct benefits to the following participants who will be involved in the EBPP processing routine and define clearly their roles in the SEP Gateway ecosystem:

- Banks: It is the entity providing the bill presentment and payment service to the end customers through its acquiring channels (ATM, POS, Internet, Mobile, etc....).
- Billers: It is the entity allowing the payment of its bills through the channels mentioned above.
 - Offline Billers: Billers without network connectivity.
 - Online Billers: Billers with network connectivity:
 - Real time: Biller does not upload the bills in SEP Database.
 - Non-Real Time: Biller uploads the bills in SEP Database.
- Electronic Bill Presentment & Payment Gateway: It is the entity allowing the single access point for billers and banks to provide the bill presentment and payment service to the end customer.

2 Bill presentment & payment workflow

2.1 Bill Upload

The bill upload is the process through which the biller sends the bills data to SEP platform to be stored in its database in order to be accessed by the end customer through bank's channels for inquiry and/or payment. This process is used by the offline billers and online/Non-Real Time billers.

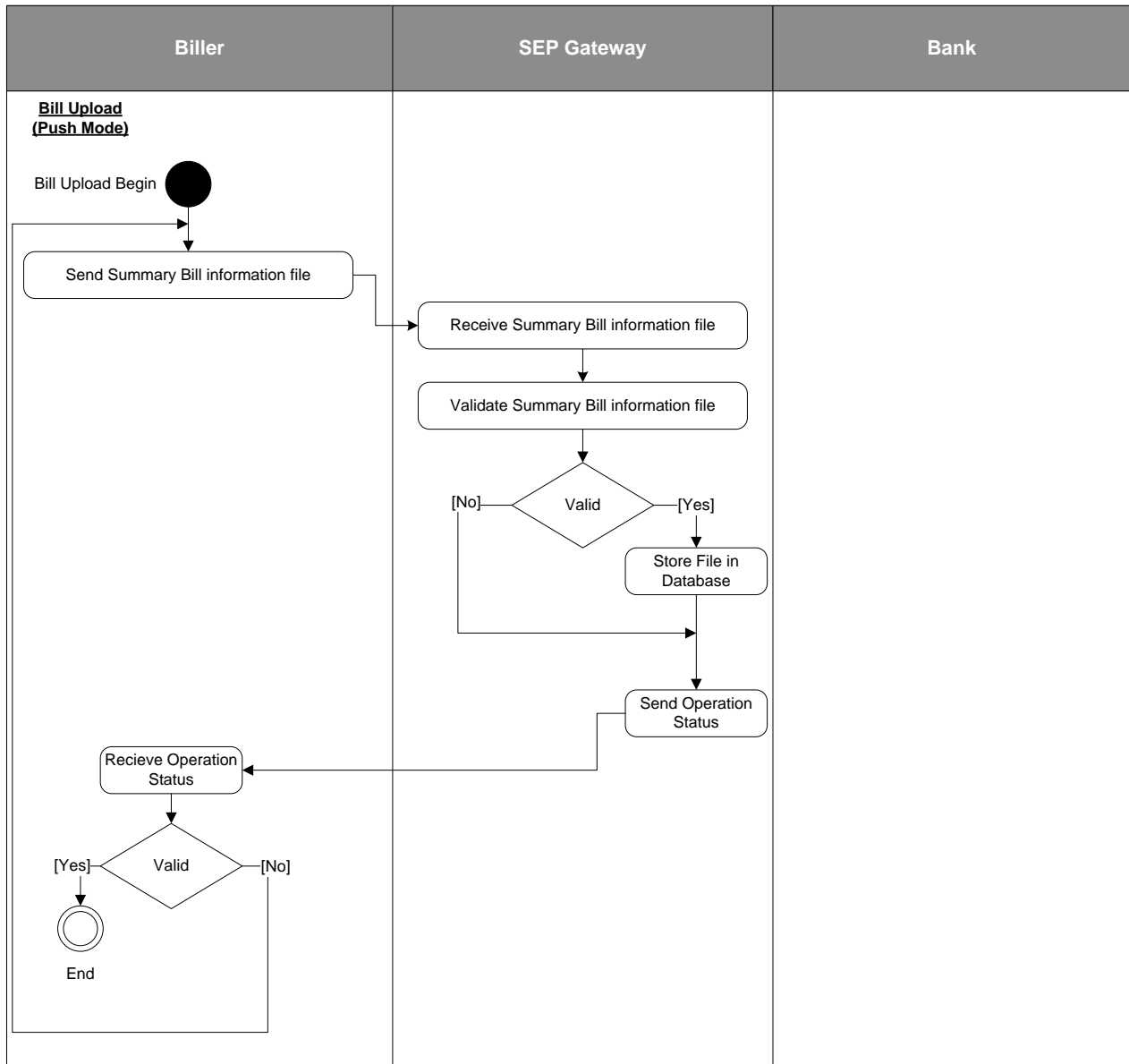
There are two modes of bill upload depending on the initiator of this operation:

- Push Mode: The biller is the initiator of the bill upload.

- Pull Mode: The biller will upload the bills upon request of SEP Gateway.

The below diagrams describe the bill upload process for the two modes.

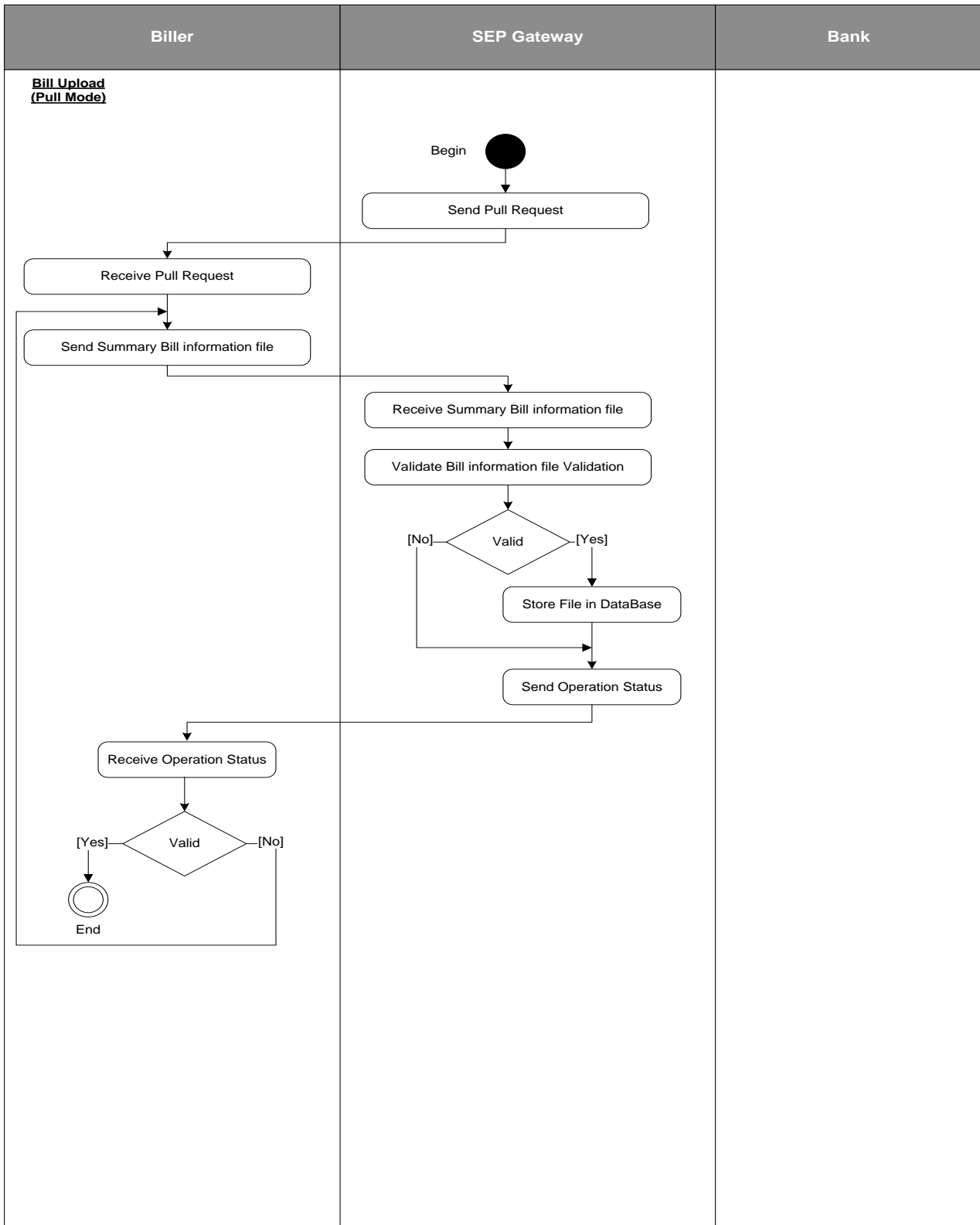
2.1.1 Bill Upload Push Mode



The previous diagram describes in general the bill upload process push mode:

-
- Biller initiates (Push) via web service using XML structure or file transfer using different formats such as XML, CSV, or any other flat file.
 - On receiving the uploaded bills, SEP Gateway performs certain validations on the bills to maintain bills data accuracy. These are:
 - Data Structure Validations.
 - Business Validations.
 - If the file/batch has errors/inconsistencies, the system rejects the entire file/batch of records and returns it to the biller for reprocessing, and it will mention the rejection reason.

2.1.2 Bill Upload Pull Mode

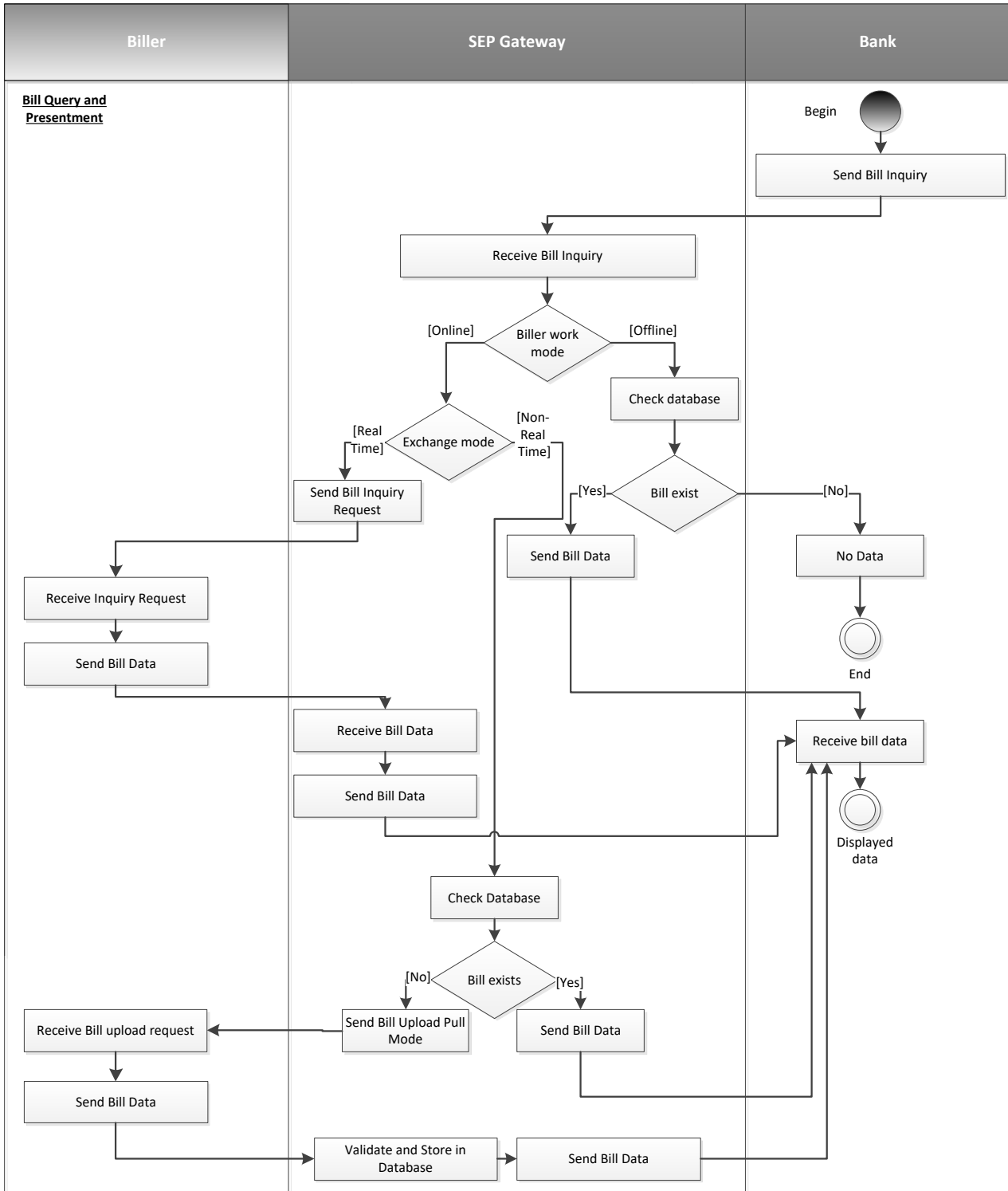


The previous diagram describes in general the bill upload process pull mode:

- SEP Gateway sends pull request via web service using XML structure.
- Biller sends the pull response informing SEP Gateway he will send bills data.
- Biller sends bills data to SEP Gateway via web service using XML file structure or file transfer using different formats such as XML, CSV, or any other flat file.
- On receiving the uploaded bills, SEP Gateway performs certain validations on the bills to maintain bills data accuracy. These are:
 - Data Structure Validations.
 - Business Validations.
- If the file/batch has errors/inconsistencies, the system rejects the entire file/batch of records and returns it to the biller for reprocessing, and it will mention the rejection reason.

2.2 Bill Presentment

The bill presentment is the process in which the bank sends a query to SEP platform about a given bill(s) to display it to the end customer on bank's channels. This process is used when the customer want to have information about a given bill(s) or prior to the bill payment. The below diagram describes the bill presentment process.



The previous diagram describes in general the bill presentment process:

- Bank applications may query SEP Gateway for bill information using a bill presentment message. The query can take the form of a Bill-Specific (single) Query in which the bank sends the appropriate parameters (Inquiry Mode, Bill No, Billing No, Service Type, IncPaid Bills) to view the data of bill(s) for specific bill number, or can take the form of Customer-Specific (multiple) Queries in which the bank sends different queries with different parameters (Inquiry Mode, Bill No, Billing No, Service Type, IncPaid Bills) in case the customer uses its customer profile to view the data of bills related to the billing numbers under its profile.
- SEP Gateway validates all the business rules (active, inactive, etc....) to be validated for each request, and based on the validation result, it either accepts or rejects the request. There are three cases to process the bill presentment request:
 - Biller Offline:
 - Upload push mode: SEP Gateway accesses to its database to extract the bill data.
 - Biller (online real-time): SEP Gateway sends bill presentment request to the biller who will send the bill data in the bill presentment response to SEP.
 - Biller (online Non-Real Time):
 - Upload push mode: SEP Gateway accesses to its database to extract the bill data.
 - Upload pull mode: SEP Gateway sends bill upload request to the biller in case the bill data does not exists in its database.
- The response of bill presentment may contain one or more records based on the criteria used in the query and might return zero result as well.

2.3 Bill Payment

The bill payment is the process in which the bank sends a payment request to SEP Gateway to validate the payment request initiated by the end customer through bank's channels. This process is used when the customer want to pay a given bill. The below diagram describes the bill payment process.

Syrian Electronic Payment Co.

Joint-Stock Company, Nominal Capital SYP 1 billion

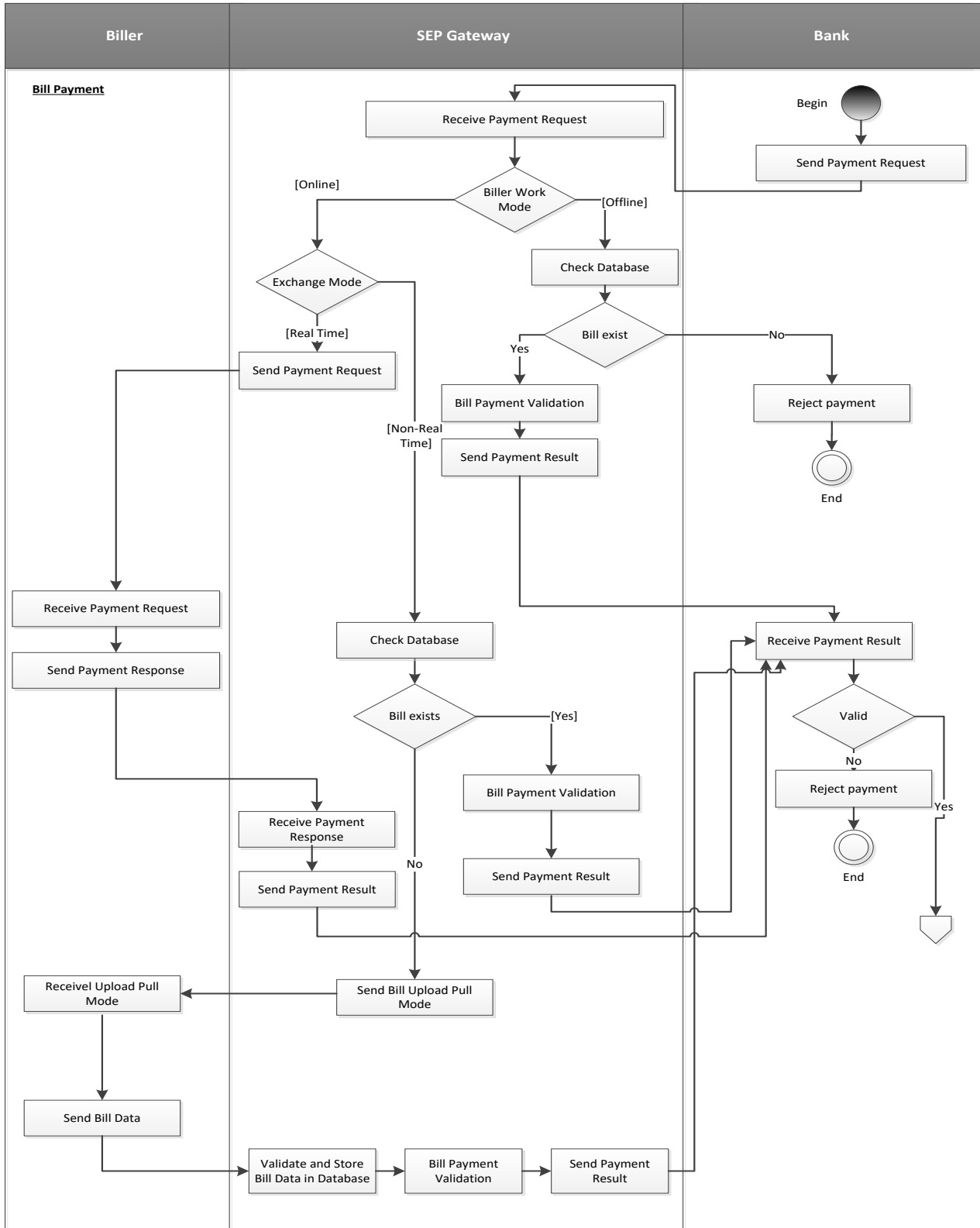
Registered under CR#17581

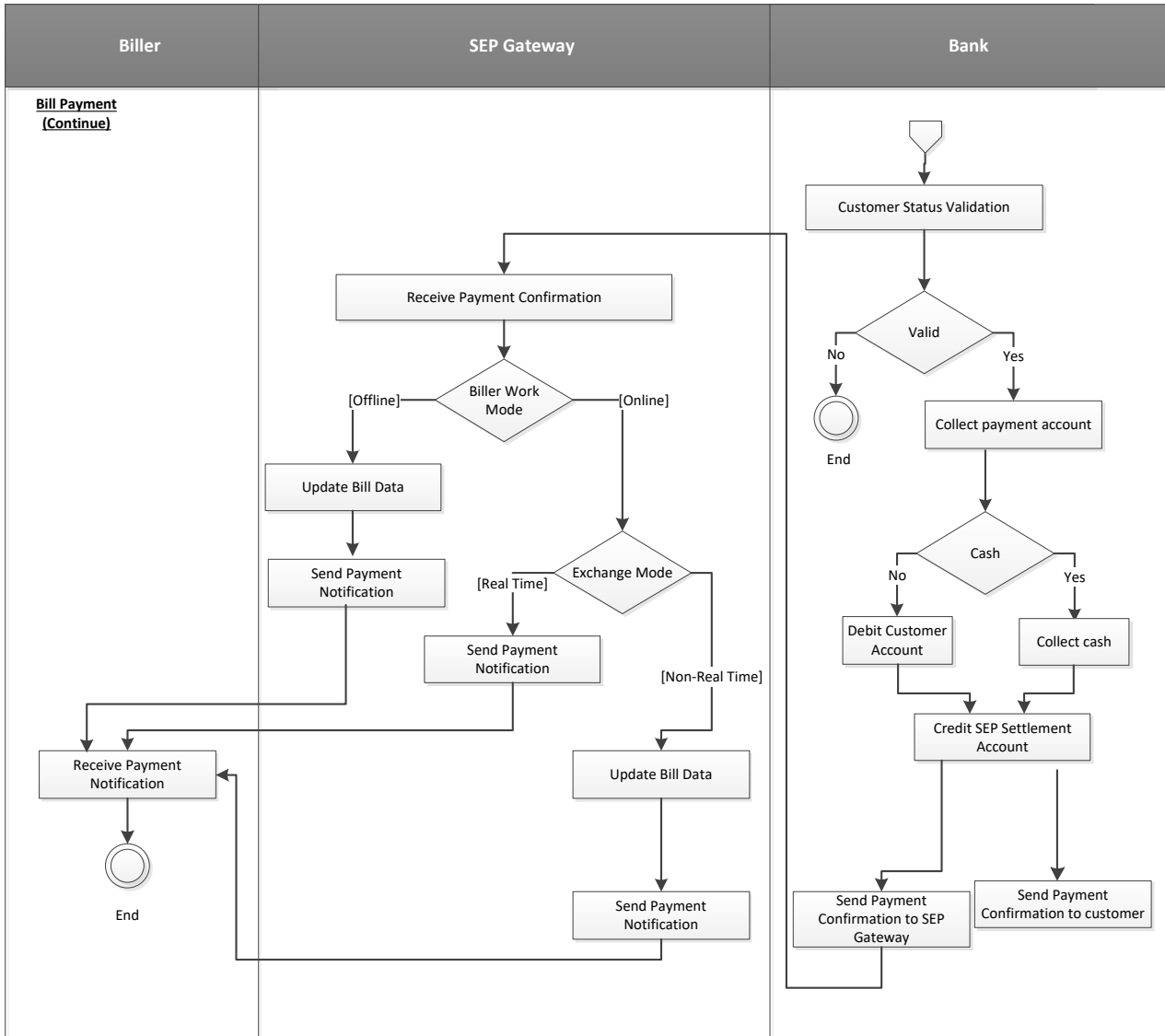
الشركة السورية للمدفوعات الإلكترونية

شركة مساهمة مغلقة رأس مالها الاسمي: مليار ليرة سورية

تملك الدولة ممثلة بالخزينة العامة أسهماها كافة

مسجلة في السجل التجاري تحت الرقم 17581





The previous diagram describes in general the bill presentment process:

- The Payment process permits Banks to create new payment records in SEP Gateway. The process is intended to ensure the customer pays according to Biller intent, and it involves a validation of Biller's payment rules.
- The bank will check the customer fund once the validation of the bill presentment request from the biller through SEP Gateway or from SEP Gateway. If the funds are sufficient, the bank should send a bill payment confirmation to SEP Gateway.
- Further to the payment confirmation received from the bank, SEP Gateway will send payment notification to the biller and will update the bill data in case the bill exists in its database. SEP must record data about all payments in storage termed as Payment Log.
- Banks must record data about all payments in storage termed as SEP Payment Log.

3 Billers Integration Process

3.1 Integration Approach

3.1.1 Integration Technology/Protocol

SEP Interfaces with billers will be based on one of the following protocols that will be used to exchange messages between billers and SEP Gateway. The interface with billers will be available based on ISO8583 or web service protocol.

3.1.1.1 ISO 8583 Protocol

ISO 8583 messages are built from the following components: message type identifier, one or two bit maps, and a series of data elements in the order of the bit map representation.

Message Length	Protocol Identification	Message Header	Message Type	Bit Maps	Data Elements
----------------	-------------------------	----------------	--------------	----------	---------------

- **Message Length:**

This field is used to identify the length of message. It should contain 4 digits used to specify the length of the message to be extracted from IP queue. This field is always required in the messages sent to or received from external interface.

▪ **Protocol Identification:**

This field is used to identify the message protocol. It should always contain the three characters “ISO”. This field is always required in the messages sent to or received from the external interface.

▪ **Message Header:**

The header is required in all the messages. The HOST should not change it. The header format is as follows:

Position	Content
01-04	Message length (including Header).
05-24	Internal Message Reference Data
25-30	Issuer Purge Time (In Seconds)

▪ **Message Type:**

▪ 1100	:	Bill Payment Request
▪ 1110	:	Bill Payment Response
▪ 1120	:	Bill Confirmation Request
▪ 1130	:	Bill Confirmation Response
▪ 1300	:	Bill Inquiry Request, Create Customer Request, Inquiry Customer Request, Add Customer Request, Remove Customer Request, Cut Off Time Information Request, Bill Upload
▪ 1310	:	Bill Inquiry Request, Create Customer Response, Inquiry Customer Response, Add Customer Response, Remove Customer Response, Cut Off Time Information Response
▪ 1804	:	Network management request,
▪ 1814	:	Network management request response.

▪ **Bit Maps:**

The ISO 8583 uses the « bit map » technique. The bit map structure shows if a data element is present (corresponding bit in the bit map = 1) or absent (corresponding bit in the bit map = 0). The bits in the bit map are numbered from left to right.

Two bit maps containing 64 bits each, can be used in the messages exchanged between SEP Gateway and the Bank:

- A primary bit map which shows the presence or absence of the fields 1 to 64.
- A secondary bit map which shows the presence or absence of the fields 65 to 128.

The primary bit map is required in every message. The secondary bit map is optional and its presence is indicated in the first bit of the primary bit map (equal to 1 means bit map 2 presents).

▪ **Data Elements:**

The third component of a message consists of a series of data elements or fields. These fields contain the actual transaction data required to process transactions across the network. Messages are constructed using the bit map as an index of the data elements that are present.

3.1.1.2 Web Service

Web service based interface is using SOAP (Simple Object Access Protocol) version 1.2 which is used for exchanging structured information in its implementation.

SOAP is based on XML as the messages format, and relies on the HTTP(S) protocols for transmission.

A SOAP message is encoded as an XML document, consisting of an <Envelope> element, which contains an optional <Header> element, and a mandatory <Body> element. The <Fault> element, contained in the <Body>, is used for reporting errors.

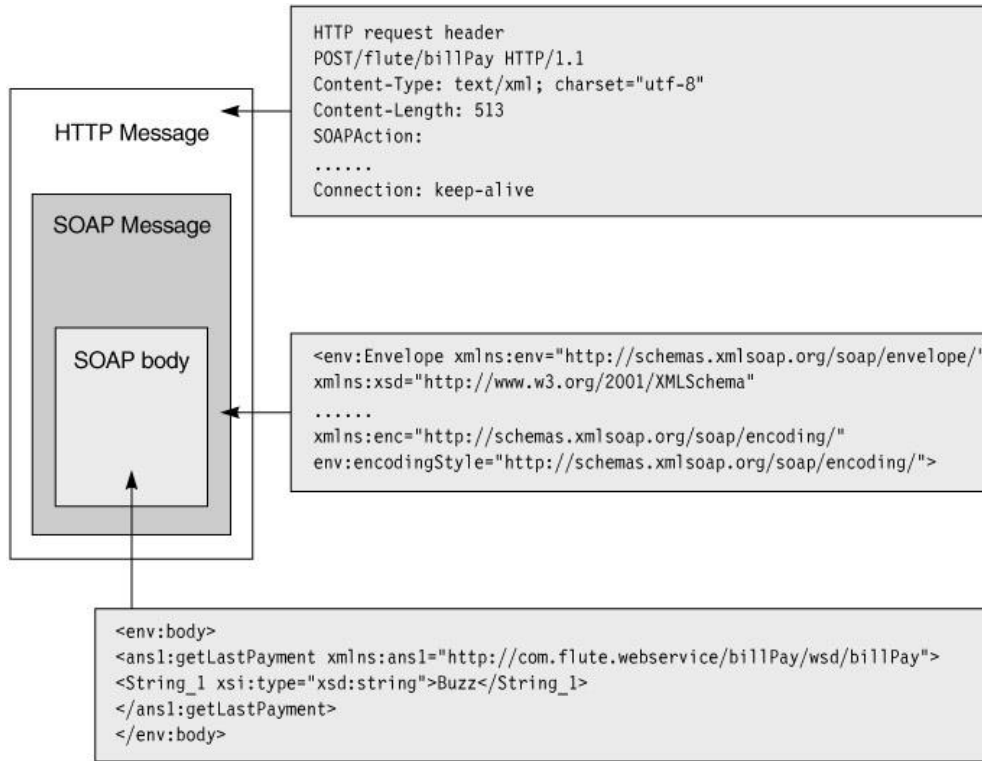
So, a SOAP message is an ordinary XML document containing the following elements:

Element	Description	Required
<Envelope>	The SOAP <Envelope> is the root element in every SOAP message. It contains two child	YES

	elements, an optional <Header>, and a mandatory <Body>.	
<Header>	The SOAP <Header> is an optional subelement of the SOAP envelope. It is used to pass application-related information that is to be processed by SOAP nodes along the message path.	NO
<Body>	The SOAP <Body> is a mandatory subelement of the SOAP envelope. It contains information intended for the ultimate recipient of the message.	YES
<Fault>	The SOAP <Fault> is a subelement of the SOAP body, which is used for reporting errors.	NO

With the exception of the <Fault> element, which is contained in the <Body> of a SOAP message, XML elements in the <Header> and the <Body> are defined by the applications that make use of them. However, the SOAP specification imposes some constraints on their structure.

The below figure shows the main elements of a SOAP message contained on a HTTP(S) Message:



3.1.2 Integration Workflow

Each bank and biller will be participant of SEP Gateway Platform should have its code and password that will be generated by SEP Gateway and communicated to the concerned party through a secured channel.

Each service, detailed in the next [paragraph 7.2](#), is published on a specific URL, where the bank or the biller will be requested to use a specific URL to invoke a certain operation or service, and only a given URL will be accessible for each bank/biller.

The format of URL is: <https://domainNameOfSEP/memberAppContext/serviceName>

- **https:** All HTTP communications between SEP Gateway and their participants will be encrypted using TLS v1.2.

- **domainNameOfSEP:** Will be a unique domain name of sub-domain specific to SEP Gateway that will be shared with participants (Billers and Banks)
- **memberAppContext:** Each participant will have its own context name, that can be only the name of this participant
- **serviceName:** Is the name of the service that should be invoked to perform one or many operations.

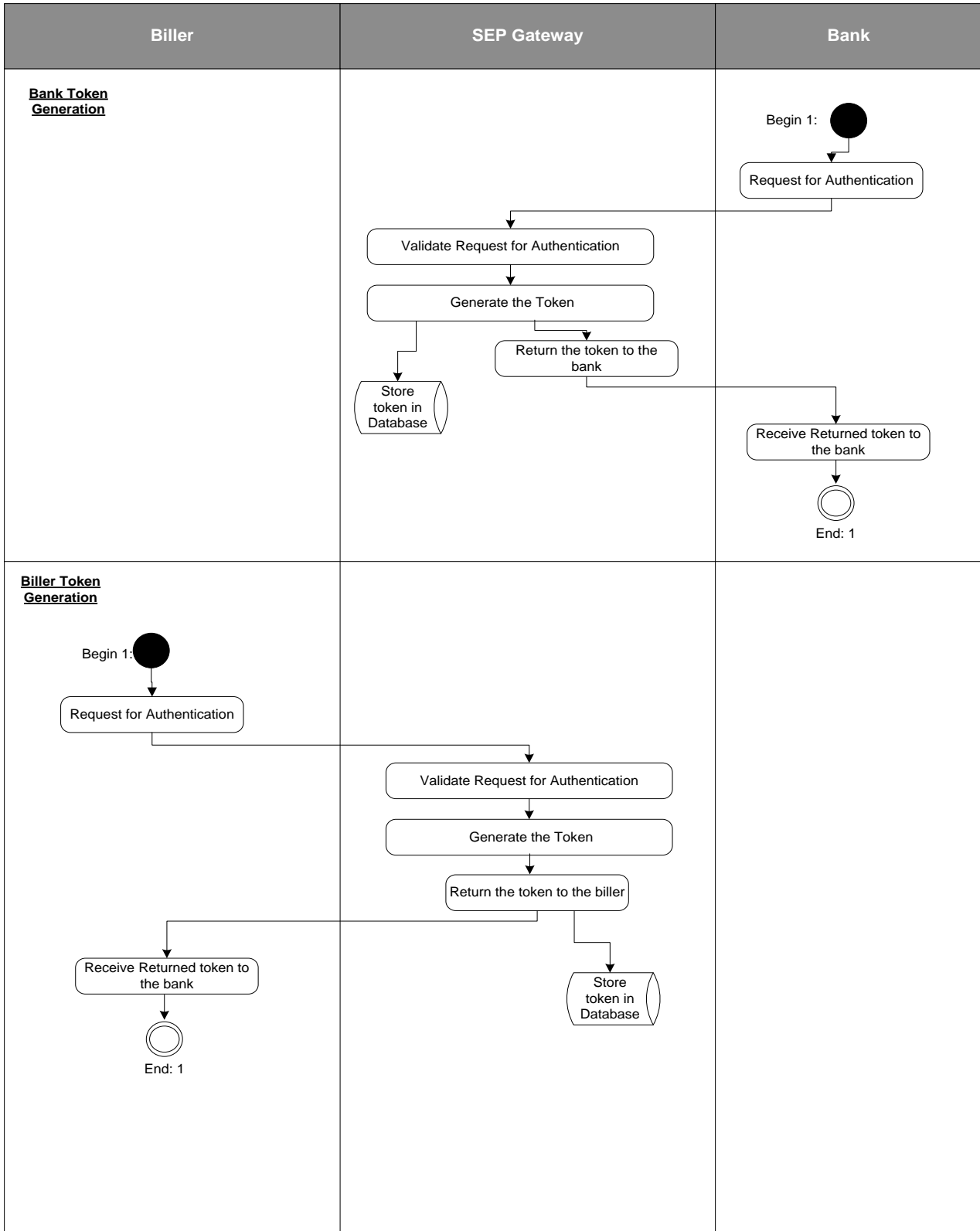
To be allowed to use the web services deployed by SEP Gateway, each bank or biller considered to be a participant of SEP, must first call the "Authentication Web Service" which will authenticate the participant and give him an access parameter to other Web Services.

Called "Token Key", this parameter is an entry ticket with a unique value that expires after a specific period. This unique information generated by SEP Gateway is a mandatory parameter to invoke the others Webservices deployed by SEP Gateway. Any call to a Web Service will be rejected directly by SEP Gateway if it does not have a valid Token.

As mentioned above, to get a token, the participant should simply invoke the authentication Webservice by presenting the following two parameters:

- Customer Code.
- Password.

The following diagram illustrates how the process of getting the token should be conducted between biller/bank and SEP Gateway:



-
- Bank/Biller call Request for Authentication.
 - SEP Gateway validates this request.
 - SEP Gateway generate the token for the requester (Biller/Bank).
 - SEP Gateway stores the token in its database.
 - SEP Gateway returns the token to the requester (Bank/Biller) for future use.

After the bank/biller gets a token, bank/biller get ready to perform any operation(Inquire about bills, pay a bill, etc...), the bank/biller has to call the correspondent web service that is responsible for performing this operation, and this requires the caller (Bank/Biller) to send the following parameters:

- Token Key: This is the token exchanged during authentication phase.
- XML Request: The request message (See interface specification).

3.1.3 Integration Security

In order to secure the exchanged data through the network, SEP Gateway has to set up the following mechanisms:

- Configures an encrypted communication between the Gateway and the participant using the TLS v1.2 protocol (Transport Layer Security) designed to establish encryption of exchanges and to guarantee the identity of the participants.
- Uses a digital signature generated using SHA-2 “SHA256withRSA”with a size of 256 bytes that will authenticate the XML message in a similar manner the handwritten signature authenticate the printed document.

This signature cannot be counterfeit and it guarantees that the initiator of the message (Biller, SEP Gateway, or Bank) has written or agreed on the content of this message to which the signature is attached.

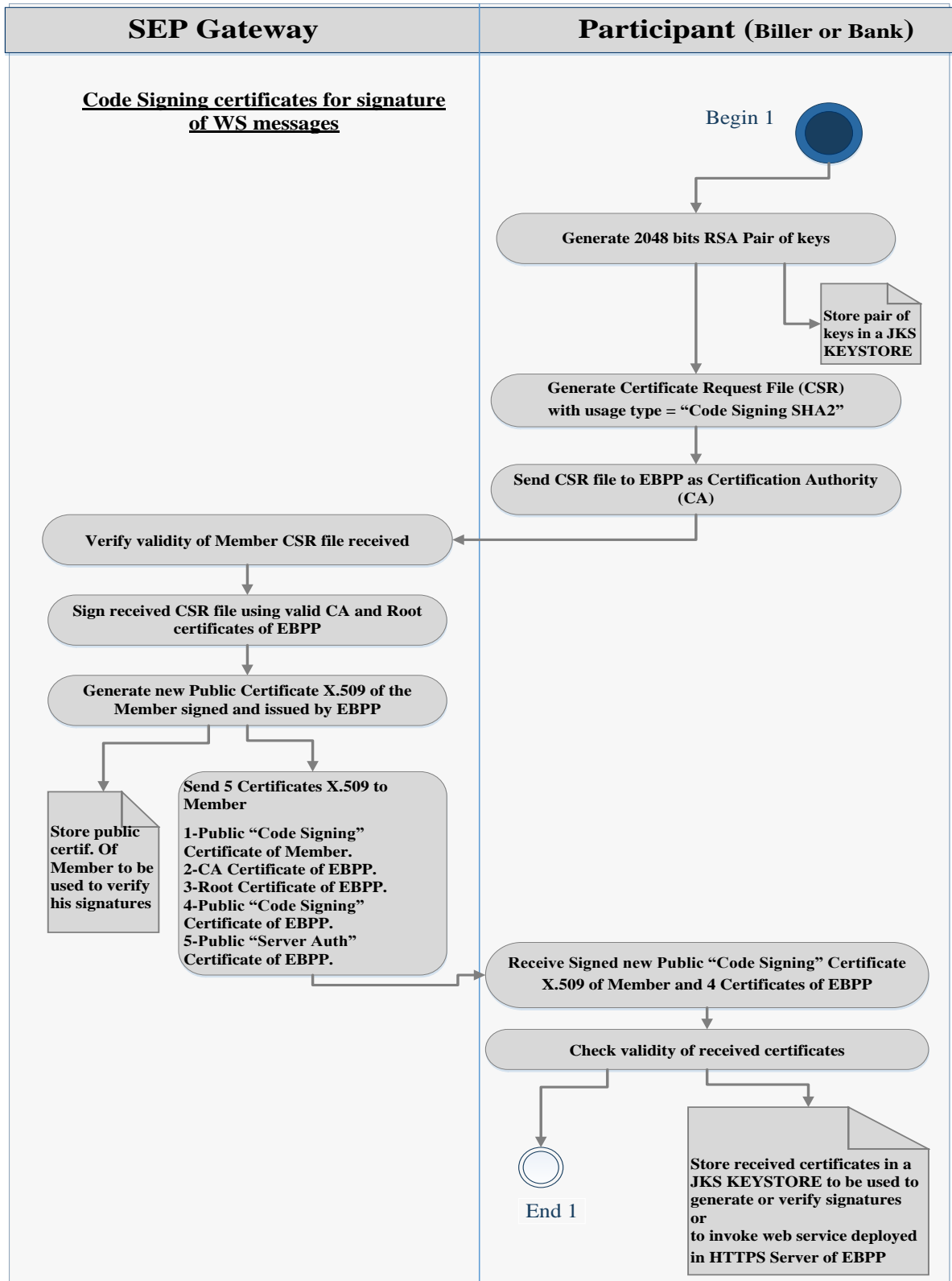
The recipient of a digitally signed message can verify that the message originated from the party whose signature is attached to the message and that the message has not been altered either intentionally or accidentally since it was signed.

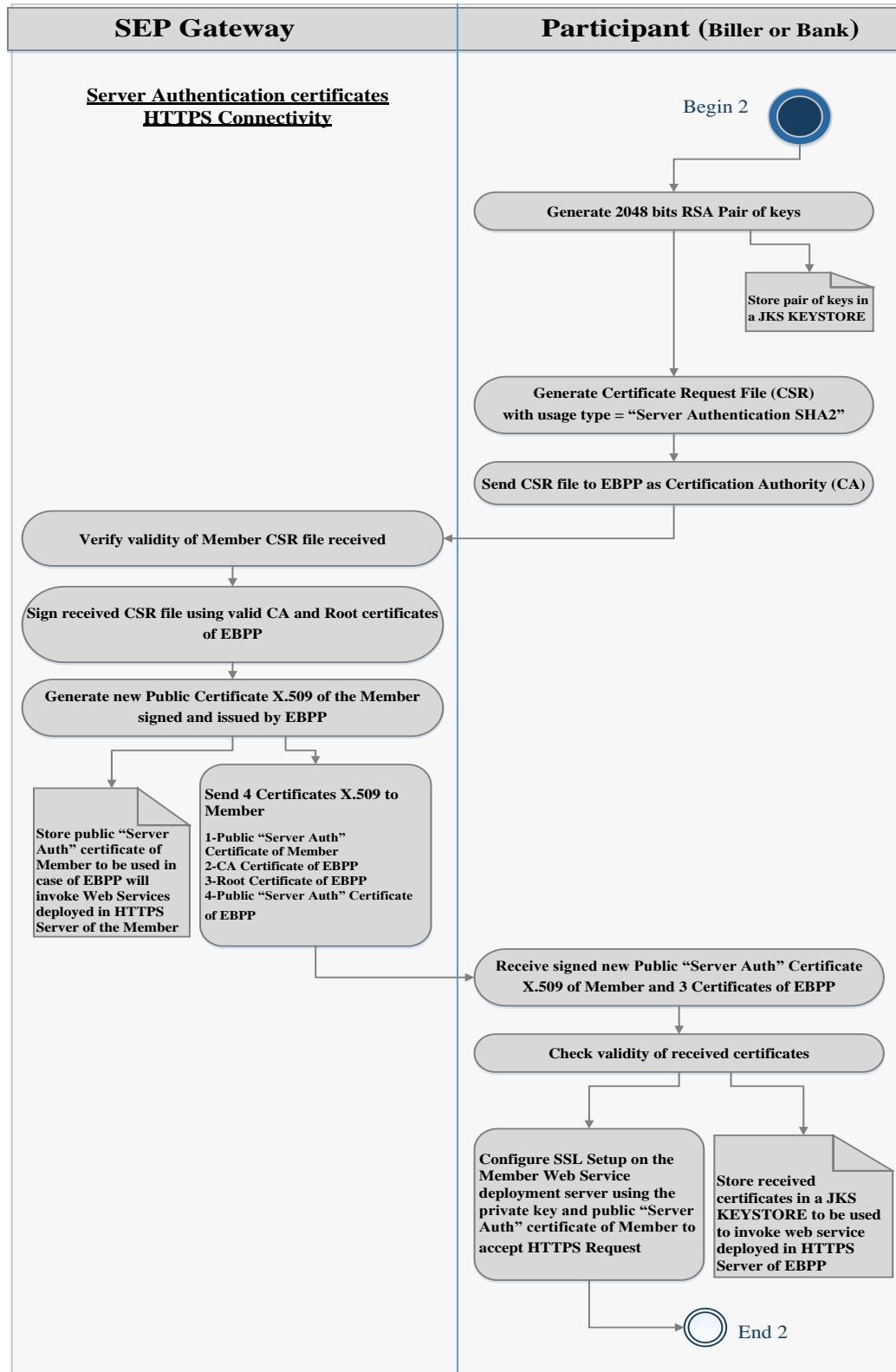
Digital signatures enable the “authentication” and “non-repudiation” of digital messages, assuring the recipient of a message of both the identity of the sender and the integrity of the message.

3.1.3.1 Getting Publics Certificates

To implement the mechanisms mentioned above, SEP proposes two options to get X.509 certificates:

Option 1: All participants will get trusted publics certificates using SEP Gateway CA.





SEP Gateway should perform the procedure of generating and exchanging X.509 certificates with each of its participants. This procedure consists of the following steps:

- The participant will generate two RSA key pairs (two Private Keys and two Public Keys) with a size of 2048 bits for both.
- The participant should then generate two certification request files based on the two RSA key pairs called CSR file 1 and CSR file 2 and communicate them to SEP Gateway (the usage type of the certificate request CSR file 1 must be "Code Signing with SHA2 signature" and that of CSR file 2 must be "Server Authentication").
- SEP Gateway will sign these CSRs file and generate two X.509 certificates authenticated by SEP Gateway as a Certificate Authority (CA):
 - One public certificate will be used for "Server Authentication".
 - The other one will be used for "Code Signing".
- SEP Gateway will communicate 5 files to the participant (Last one is optional):
 - Public certificate of the participant for server authentication.
 - Public certificate of the participant for code signing.
 - Public certificate of SEP Gateway.
 - Public CA certificate of SEP Gateway.
 - Public root certificate of SEP Gateway (optional) used to sign the CA certificate of SEP.

Option 2: Each participant will get public certificates trusted by foreign Certificate Authority

This procedure consists of the following steps:

- The participant will generate two RSA key pairs (two Private Keys and two Public Keys) with a minimum size of 2048 bits for both.
- The participant should then generate two certification request files based on the two RSA key pairs called CSR file 1 and CSR file 2 and communicate them to a trusted Certificate Authority (the usage type of the certificate request CSR file 1 must be "Code Signing with SHA2 signature" and that of CSR file 2 must be "Server Authentication").

- The trusted Certificate Authority will sign these CSRs file and generate 2 X.509 certificates authenticated by the CA Certificate of this Trusted Certificate Authority:
 - One public certificate will be used for “Server Authentication”.
 - The other one will be used for “Code Signing”.
- The trusted Certificate Authority will communicate three or four files to the participant:
 - Public certificate of the participant for server authentication.
 - Public certificate of the participant for code signing.
 - Public CA certificate of the trusted Certificate Authority.
 - Public Root certificate of the trusted certificate authority (This is an option file and depends on each certificate authority).
- The participant will communicate to SEP Gateway three or four files:
 - Public certificate of the participant for server authentication.
 - Public certificate of the participant for code signing.
 - Public CA certificate of trusted Certificate Authority.
 - Public Root certificate of the trusted certificate authority (This is an option file and depends on each certificate authority).

This is a non-exclusive list of Trusted Certificate Authority:

(Digicert, VISA, MicroSec, Entrust.net, VeriSign Inc. , GlobalSign, GoDaddy.com, SSL Corporation, Swisscom, Symantec Corporation, TrustCor System,).

SEP Gateway will get publics certificates trusted by another Certificate Authority:

SEP Gateway must follow same above procedure (Option 2) used by participant to get their two public certificates (listed below). After getting both certificates (Server Auth + Code Signing), SEP Gateway will share with all participants the following certificates:

- Public certificate of SEP for server authentication.
- Public certificate of SEP for code signing.

Syrian Electronic Payment Co.

Joint-Stock Company, Nominal Capital SYP 1 billion

Registered under CR#17581

الشركة السورية للمدفوعات الإلكترونية

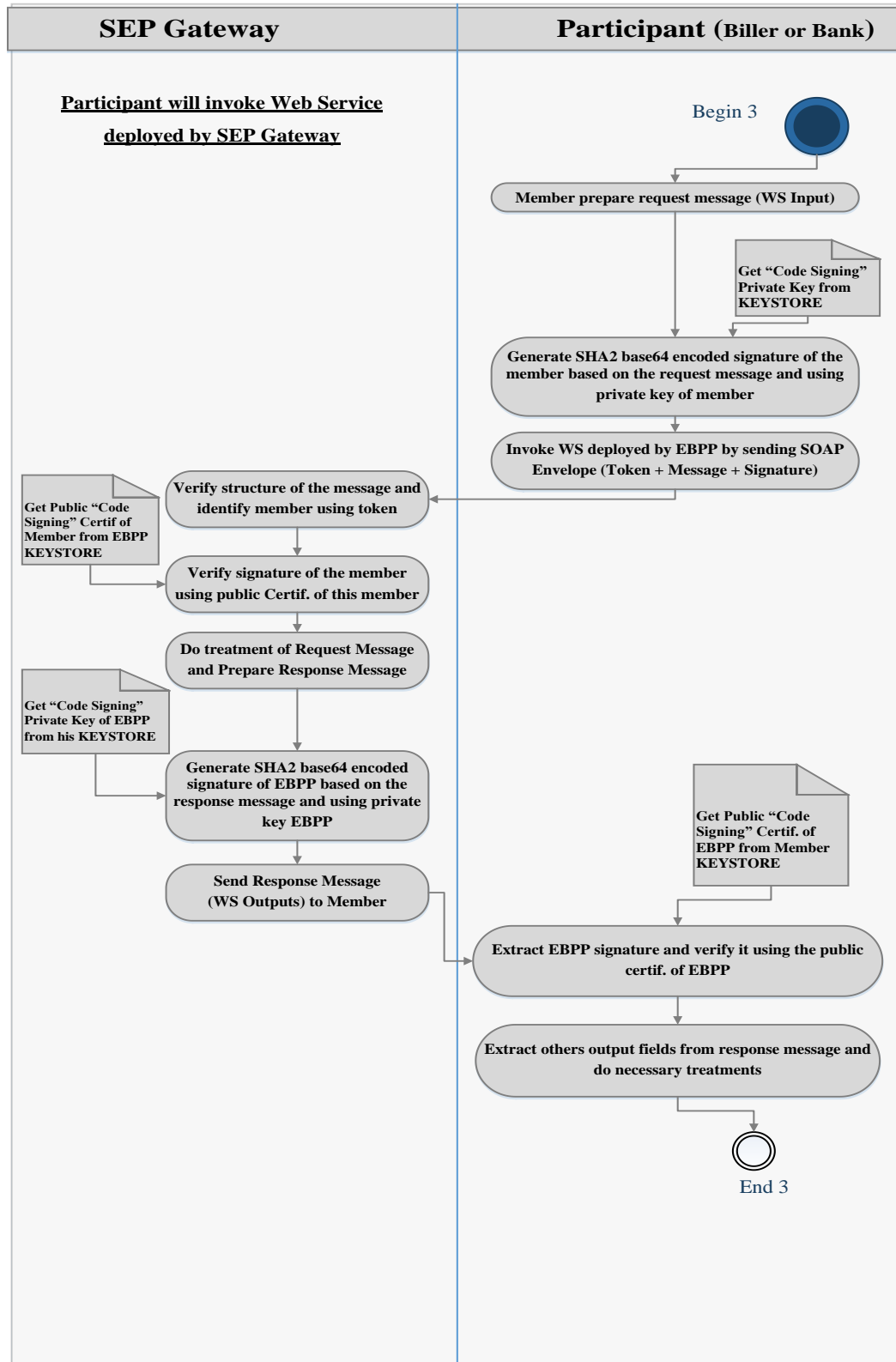
شركة مساهمة مغلقة رأسمالها الاسمي: مليار ليرة سورية

تملك الدولة ممثلة بالخرزينة العامة أسهمها كافة

مسجلة في السجل التجاري تحت الرقم 17581

3.1.3.2 Invoke Web Services

Participant invokes Web Service deployed by SEP Gateway



Once the participant holds his two public certificates and holds also the two public certificates of SEP, the participant will be able to reach HTTPS Server of SEP Gateway through the Server Certificate of SEP, and will be also able to send messages to SEP and attach participant signature to this request message.

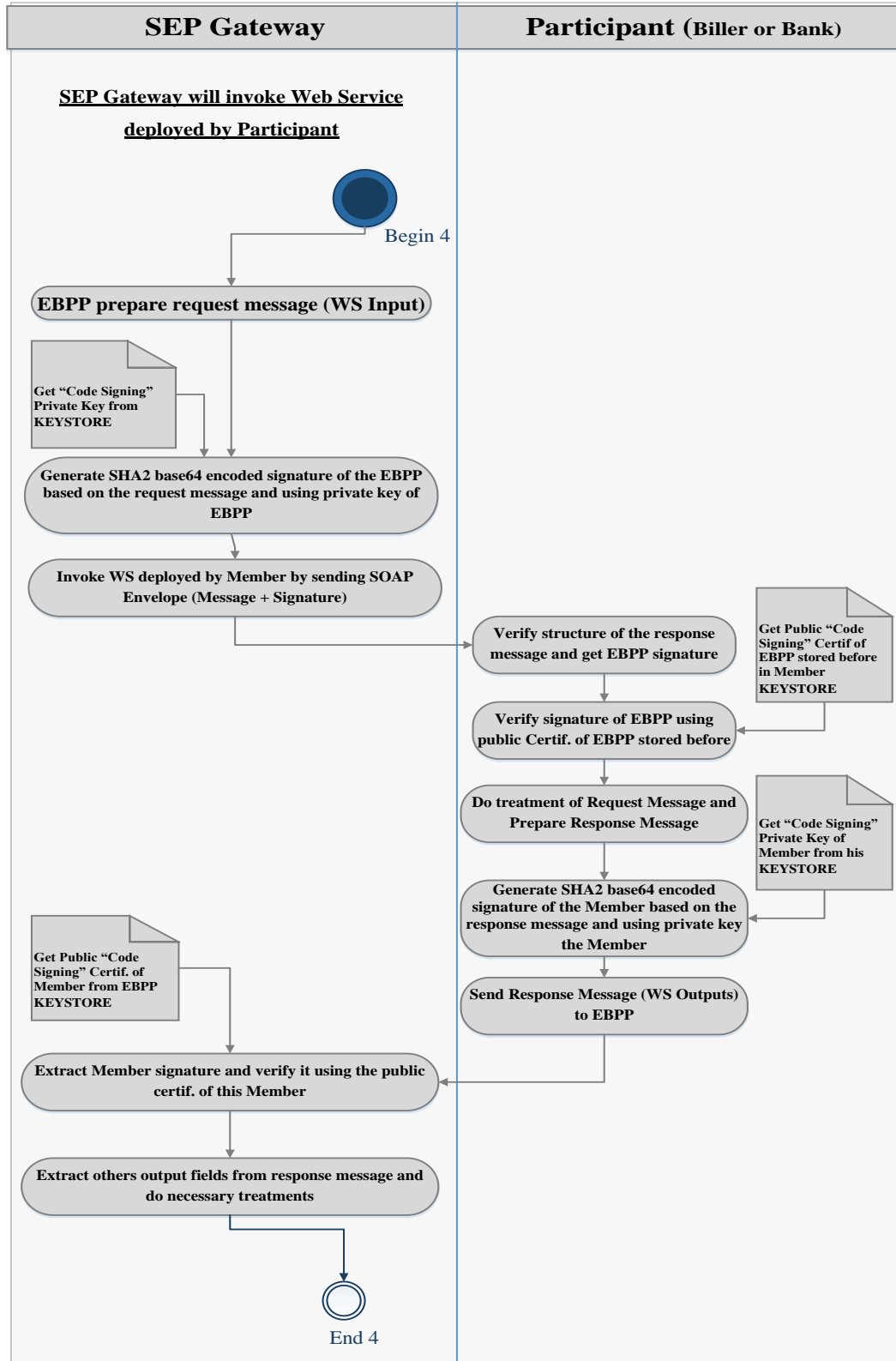
The participant must generate the signature using its private key.

Upon receipt of this request message, SEP Gateway will verify the signature using the public certificate of the participant and ensure the integrity of the message and the authenticity of the participant.

SEP Gateway will prepare the response message, and generate his proper signature using the private key of SEP and send it to the participant.

Upon receipt of this response message, the participant will verify the signature using the public certificate of the SEP Gateway and ensure the integrity of the message and the authenticity of SEP Gateway.

SEP Gateway invokes Web Service deployed by Participant



Once the participant holds his two public certificates and holds also the two public certificates of SEP, the participant should first secure his server with the private/public certificates used for “Server Authentication”. This will allow SEP Gateway to communicate with the participant and invoke their web services securely.

Once the HTTPS Server of participant is ready, SEP Gateway will be able to reach this HTTPS Server through the Server Certificate of this participant, and will be also able to send messages to the participant and attach SEP signature to this request message.

SEP Gateway must generate the signature using its private key.

Upon receipt of this request message, the participant will verify the signature using the public certificate of SEP Gateway and ensure the integrity of the message and the authenticity of SEP Gateway.

The participant will prepare the response message, and generate his proper signature using the private key of the participant and send it to SEP Gateway.

Upon receipt of this response message, SEP Gateway will verify the signature using the public certificate of the participant and ensure the integrity of the message and the authenticity of the participant.

3.1.4 Enumerated Types

SEP Gateway solution uses several enumerated types to make sure that the system is consistent and dynamic at the same time by forcing communication protocols to have certain values, where these values are used to determine the solution behavior, do certain calculations and validations, and bring up related results accordingly.

Enumerated types are used as parameters in the web services requests and are validated by SEP Gateway platform in each request/response.

3.1.4.1 Biller Categories

The category to which the billers should be linked, such as: Telecommunications, Electricity, etc...

The intent is to permit customers to query on bills according to the type of service for which they are being charged. The actual biller category is determined by participating Billers, but must be registered and approved by SEP Gateway before it is accepted. (For example)

Code	Description
GOVT	Government Service
TELC	Telecommunication
INSR	Insurance
UTIL	Utility
TRAN	Transportation

3.1.4.2 Service Types

A Service type assigns behavior to a bill, thus, permitting SEP Gateway to apply rules to bills in various manners. Billers must assign values of their choosing to each of their bills, but must first register these strings with SEP Gateway to avoid rejection. (For example)

Code	Description
FixedLine	A fixed Line
Internet	Internet
Mobile	Mobile
Electricity	Electricity

3.1.4.3 Pre-Paid Denomination

This enumerator describes the possible denomination of prepaid service type.

3.1.4.4 Bill Types

This enumerator defines the possible types of the bill being paid in terms of recurrence.

Code	Description
Recurring	Represents a normal recurring bill
OneOff	Represents a one-off bill

3.1.4.5 Payment Statuses

This enumerator describes the possible payment statuses.

Code	Description
PmtInProg	New Payment Request validated by SEP Gateway and not yet confirmed by the bank
PmtNew	New Payment Request validated by SEP Gateway and confirmed by the bank
PmtCanc	New Payment Request validated by SEP Gateway and rejected by the bank
PmtUNSet	During the reconciliation process, if a “PmtNew” transaction is matched then it will be updated to “PmtUNSet”
PmtSet	During the settlement process, if a “PmtUNSet” transaction is settled correctly in the system, then the status will be updated to “PmtSet”

3.1.4.6 Bill Statuses

The Bill Status identifies the bill’s statuses within SEP Gateway and is intended to be used when communicating bill data between separated systems.

Code	Description
BillNew	Bill record is new, no corresponding bill exists in the System
BillUpdated	Bill is an updated version of a bill currently in the system
BillPaid	The bill is fully paid and the due amount is set to zero
BillPartialPd	One or more payments have been applied to the bill, but an outstanding amount remains
BillOverPd	The bill has the paid amount greater than the due amount: the paid amount can consist of one or multiple payments.
BillExpired	Bill is inquirable and no longer payable
BillClosed	Bill is neither inquirable nor payable

These statuses will be managed in SEP system according to the following rules:

- In “Thin” mode for which the biller sends the summary about a given billing number. first time the the biller upload the record about this billing number in SEP Gateway will have the status “BillNew”, every time a new record for the same billing number is uploaded in SEP Database will have the status “BillUpdated”.
- In “Thick” mode for which the biller sends the bills for a given billing number. First time the biller upload the record about this bill number under the billing number in SEP Gateway will have the

status “BillNew”, every time a new record for the same bill number under billing number is uploaded in SEP Database will have the status “BillUpdated”.

- If a given bill represented by billing number (Thin mode) or bill number (Thick mode) is fully paid (the status of the payment should be “PmtUNSet” or “PmtSet” in order to impact the status of the corresponding bill) its status will be changed to “BillPaid” in case its current status is “BillNew” or “BillUpdated” or “BillPartialPd”.
- If a given bill represented by billing number (Thin mode) or bill number (Thick mode) is partially paid (the status of the payment should be “PmtUNSet” or “PmtSet” in order to impact the status of the corresponding bill) its status will be changed to “BillPartialPd” in case its current status is “BillNew” or “BillUpdated”.
- If a given bill represented by billing number (Thin mode) or bill number (Thick mode) is over paid (the status of the payment should be “PmtUNSet” or “PmtSet” in order to impact the status of the corresponding bill) its status will be changed to “BillOverPd” whatever its current status is “BillNew” or “BillUpdated” or “BillPartialPd” or “BillPaid”.
- The issue date, open date and due date do not impact the status of the bill.
- If the expiry date of the bill is reached its status will be changed to “BillExpired” in case its current status is “BillNew” or “BillUpdated” or “BillPartialPd”
- If the close date of the bill is reached its status will be changed to “BillClosed” in case its current status is “BillNew” or “BillUpdated” or “BillExpired” or “BillPartialPd”.

3.1.4.7 Access Channels

The access channel is a channel where customers who owe bills used to pay through, these channels are bank provided. (For example)

Code	Description
ATM	Bank Automated Teller Machine
KIOSK	Bank Kiosk
INTERNET	Internet
BTELLER	Bank Teller
POS	Point Of Sale
CCC	Call Center

Mobile	Mobile Phone
--------	--------------

3.1.4.8 Payment Methods

This enumerator describes the possible payment methods used for paying bills, which means that a bill can be paid as cash, by bank account, credit card, etc..... (For example)

Code	Description
CASH	Cash
CCARD	Credit Card
ACTDEB	Account Debit
WALLET	WALLET

3.1.4.9 Payment Types

This enumerator describes the possible payment types in terms of collection period. (For example)

Code	Description
Yearly	Payment is collected on Yearly basis
Monthly	Payment is collected on Monthly basis
One-shot	Payment is collected one-shot

3.1.4.10 Billing Number Statuses

This enumerator describes the possible statuses of the specific Billing number.

Code	Description
Active	Billing Account is Active
InActive	Billing Account is InActive
Blocked	Billing Account is Blocked

3.1.4.11 Official ID Types

This enumerator describes the possible values for the ID Types. (For example)

Code	Description
NAT	National ID
PAS	Passport ID
CIS	Commercial ID

3.1.4.12 Customer Profile Nationalities

This enumerator describes the possible nationalities of customers. (For example)

Code	Description
SY	SYRIAN

3.1.4.13 Payment Currencies

This enumerator describes the possible currencies used in SEP Gateway. (For example)

Code	Description
SYL	SYRIAN POUND

4 SOAP REQUEST and RESPONSE STRUCTURE

4.1 WS Request Structure

All SOAP envelopes for “input” requests of the web services presented below, must respect the following structure:

Request Envelope Structure	Description
<pre><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:bil="http://service/Web_service_name.wsdl" xmlns:typ="http://service/ Web_service_name.wsdl/types/"></pre>	<p>Start of request SOAP envelope:</p> <ul style="list-style-type: none"> - The "Web_service_name" Represents the name of the web service with the first character in uppercase. - Example "Biller_bill_presentment". - This envelope must present 3 mandatories definitions of namespaces “soapenv”, "bil" and "typ" (see details on WSDL files shared by SEP)
<pre><soapenv:Header/></pre>	<ul style="list-style-type: none"> - Header parameters will not be use on SEP Gateway Context
<pre><soapenv:Body></pre>	<ul style="list-style-type: none"> - Start of body
<pre><bil:WebServiceOperationName></pre>	<ul style="list-style-type: none"> - SOAP Action for the called operation based on web service name
<pre><wsRequest></pre>	<ul style="list-style-type: none"> - All request must have 2 mandatories nodes <paramIn> & <signature>
<pre><typ:paramIn></pre>	<ul style="list-style-type: none"> - XML presentation of all below described input parameters

<pre><typ:param1>value</typ:param1> <typ:param2>value</typ:param2></pre>	- List on input paramaters
<pre><typ:paramX> <typ:array> <typ:paramX_1>...</typ:paramX_1> <typ:paramX_2>...</typ:paramX_2> ... <typ:paramX_n>...</typ:paramX_n> </typ:array> <typ:array> <typ:paramX_1>...</typ:paramX_1> <typ:paramX_2>...</typ:paramX_2> ... <typ:paramX_n>...</typ:paramX_n> </typ:array> </typ:paramX></pre>	- In case of input array container, it will be necessary to repeater the tag <array> as many times as count of records contained in the array container of the type "paramX" for example.
<pre><typ:paramY> <typ:paramY_1>...</typ:paramY_1> <typ:paramY_2>...</typ:paramY_2> ... <typ:paramY_n>...</typ:paramY_n> </typ:paramY></pre>	- In case of input container, we will have to present the sub elements between the 2 tags <typ: paramY> and </typ:paramY>
<pre></typ:paramIn></pre>	- End of input paramater
<pre><typ:signature></typ:signature></pre>	- Base64 encoded format of the generated signature of SEP or Biller. See chapter 3.1 for more details
<pre></wsRequest></pre>	
<pre></bil:WebServiceOperationName></pre>	- End of SOAP Action
<pre></soapenv:Body></pre>	- End of body
<pre></soapenv:Envelope></pre>	- End of envelope

Subsequently, all input parameters specified below for each web services be must be presented in XML format encapsulated by 2 XML tags <typ:paramIn> and </typ: paramIn>.

4.2 WS Response Structure

Syrian Electronic Payment Co.

Joint-Stock Company, Nominal Capital SYP 1 billion

Registered under CR#17581

الشركة السورية للمدفوعات الإلكترونية

شركة مساهمة مغلقة رأسمالها الاسمي: مليار ليرة سورية

تملك الدولة ممثلة بالخزينة العامة أسهمها كافة
مسجلة في السجل التجاري تحت الرقم 17581

All SOAP envelopes for “output” responses of the web services presented below, must respect the following structure:

Request Envelope Structure	Description
<code><env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"></code>	Start of response SOAP envelope: - This envelope must present the definition of namespace “env”
<code><env:Header/></code>	- Header parameters will not be use on SEP Gateway Context
<code><env:Body></code>	- Start of body
<code><m:WebServiceOperationNameResponse xmlns:m="http://service/ Web_service_name.wsdl"></code>	- SOAP Action based on web service name with prefix “Response” - it present at least the definition of namespace “m”
<code><result></code>	- All response ‘result’ must have 2 mandatories nodes <code><paramOut></code> & <code><signature></code>
<code><typ:paramOut xmlns:typ="http://service/ Web_service_name.wsdl/types/"></code>	- XML presentation of all below described output parameters - it present also the definition of namespace “typ”
<code><typ:param1>value</typ:param1></code> <code><typ:param2>value</typ:param2></code>	- List on output paramaters
<code><typ:paramX></code> <code><typ:array></code> <code><typ:paramX_1>...</typ:paramX_1></code> <code><typ:paramX_2>...</typ:paramX_2></code> ... <code><typ:paramX_n>...</typ:paramX_n></code> <code></typ:array></code> <code><typ:array></code> <code><typ:paramX_1>...</typ:paramX_1></code> <code><typ:paramX_2>...</typ:paramX_2></code> ... <code><typ:paramX_n>...</typ:paramX_n></code> <code></typ:array></code> <code></typ:paramX></code>	- In case of output array container, it will be necessary to repeater the tag <code><array></code> as many times as count of records contained in the array container of the type "paramX" for example.

<pre> <typ:paramY> <typ:paramY_1>...</typ:paramY_1> <typ:paramY_2>...</typ:paramY_2> ... <typ:paramY_n>...</typ:paramY_n> </typ:paramY> </pre>	- In case of output container, we will have to present the sub elements between the 2 tags <typ: paramY> and </ typ:paramY>
<pre> </typ:paramOut> </pre>	- End of output paramaters
<pre> <typ:signature xmlns:typ="http://service/Web_service_name.wsdl/types/"> </typ:signature> </pre>	- Base64 encoded format of the generated signature of SEP or Biller. See chapter 3.3 for more details - it present also the definition of namespace “typ”
<pre> </result> </pre>	
<pre> </m: WebServiceOperationNameResponse > </pre>	- End of SOAP Action
<pre> </env:Body> </pre>	- End of body
<pre> </env:Envelope> </pre>	- End of envelope

Subsequently, all output parameters specified below for each web services be must be presented in XML format encapsulated by 2 XML tags <typ:paramOut> and </typ:paramOut>.

5 Digital signature management

5.1 Generation of signature for request message

Once the input request is prepared and ready to be sent by the “Sender” who invoke the WS, the sender must sign the input request before sending it to “Receiver” who deploy the WS, and must necessary follow the below generation procedure:

<p>Step 1 : Take only the content of the 2 tags <typ: paramIn> and </ typ: paramIn> from SOAP request envelope</p>		
<pre> <typ:paramIn> <typ:param1>..</typ:param1> <typ:paramY> <typ:paramY_1>...</typ:paramY_1> <typ:paramY_2>...</typ:paramY_2> ... </pre>	<pre> => </pre>	<pre> <typ:param1>..</typ:param1> <typ:paramY> <typ:paramY_1>...</typ:paramY_1> <typ:paramY_2>...</typ:paramY_2> ... </pre>

Syrian Electronic Payment Co.

Joint-Stock Company, Nominal Capital SYP 1 billion

Registered under CR#17581

الشركة السورية للمدفوعات الإلكترونية

شركة مساهمة مغلقة رأس مالها الاسمي: مليار ليرة سورية

تملك الدولة ممثلة بالخزينة العامة أسهماها كافة
مسجلة في السجل التجاري تحت الرقم 17581

<pre><typ:paramY_n>...</typ:paramY_n> </typ:paramY> ... <typ:paramN>xx</typ:paramN> </typ:paramIn></pre>		<pre><typ:paramY_n>...</typ:paramY_n> </typ:paramY> ... <typ:paramN>xx</typ:paramN></pre>
<p>Step2: Remove all “space”, “return line”, “blank line” and “tabulation”.</p>		
<pre><typ:param1>..</typ:param1> <typ:paramY> <typ:paramY_1>...</typ:paramY_1> <typ:paramY_2>...</typ:paramY_2> ... <typ:paramY_n>...</typ:paramY_n> </typ:paramY> ... <typ:paramN>xx</typ:paramN></pre>	<p>==></p>	<pre><typ:param1>..</typ:param1> <typ:paramY> <typ:paramY_1> ... </typ:paramY_1> <typ:paramY_2> ... </typ:paramY_2> ... <typ:paramY_n> ... </typ:paramY_n> </typ:paramY> ...<typ:paramN> .. </typ:paramN></pre>
<p>STEP 3: Afterwards, a byte-representation of input request is generated via UTF-8 encoding</p>		
<p>STEP 4: A digital signature is then generated using senders (SEP or Biller) signing RSA private key for the just created byte-representation via SHA2withRSA hashing algorithm.</p>		
<p><i>UTF8 byte representation</i></p>	<p>==></p>	<pre>085def4766beb5f5491fc4b174ba2077e3 eb41b382ccbd424016d0995d8d2d21cdf5 d590b8ae346b69a4858b9e963d9bec4fc6 c126c3fe538e05a03fd57562a4128897 98f82ade1438d406f3c0291881542dca9 977c4447e39bf3e474d4a032833f5908be e0b9767c2b774f777fcf0a94ee372e6f513 9a45ad17dba1659e82a7bebea66998636b 93656d1a8d00bb9a6fbc488dc8c8db92f7 e49c831f14316b1d8f02146a5522f37677 f255d09aabad515cff34a722aea3c330dc0 8cf9c3a0c0a46920465ad466a4e1a165fd 52e40991a17a62de2a92c5a4e5a50510b3</pre>

		ce50ebaf98501440420fea6a640c4ae2cbfc50d3ab90f4be5cbfd22a39b249b67672
<p>STEP 5: Finally, the digital signature is generated through encoding the hash result to Base64String.</p>		
<pre>085def4766beb5f5491fc4b174ba2077e3eb41b3 82ccbd424016d0995d8d2d21cdf5d590b8ae346b 69a4858b9e963d9bec4fc6c126c3fe538e05a03fd fd57562a412889798f82ade1438d406f3c029188 1542dcda9977c4447e39bf3e474d4a032833f590 8bee0b9767c2b774f777fcf0a94ee372e6f5139a4 5ad17dba1659e82a7bebea66998636b93656d1a8 d00bb9a6fbc488dc8c8db92f7e49c831f14316b1 d8f02146a5522f37677f255d09aabad515cff34a7 22aea3c330dc08cf9c3a0c0a46920465ad466a4e1 a165fd52e40991a17a62de2a92c5a4e5a50510b3 ce50ebaf98501440420fea6a640c4ae2cbfc50d3a b90f4be5cbfd22a39b249b67672</pre>	==>	<pre>CF3vR2a+tfVJH8SxdLogd+PrQbOCzL 1CQBbQmV2NLSHN9dWQuK40a2mkh Yuelj2b7E/GwSbD/IOO BaA/39V1YqQSiJeY+CreFDjUBvPAK RiBVC3NqZd8REfjm/PkdNSgMoM/W Qi+4Ll2fCt3T3d/zwqU 7jcub1E5pFrRfboWWegqe+VqZpmGNr k2VtGo0Au5pvvEiNyMjBkvfknIMfFDF rHY8CFGpVIvN2d/ JV0JqrrVFc/zSnIq6jwzDcCM+cOgwKR pIEZa1Gak4aFl/VLkCZGhemLeKpLFp OWIBRCzzlDrr5hQ FEBCD+pqZAxK4sv8UNOrkPS+XL/S KjmySbZ2cg==</pre>

5.2 Verify the signature of the request message

Once the input request is received by the “Receiver”, this one must verify the signature of the “Sender” before processing the message and the receiver must necessary follow the below verification procedure:

<p>STEP 1: The received Base64String signature of sender must be decoded to byte array format.</p>		
<pre>CF3vR2a+tfVJH8SxdLogd+PrQbOCzL1CQBbQmV 2NLSHN9dWQuK40a2mkhYuelj2b7E/GwSbD/IOO BaA/39V1YqQSiJeY+CreFDjUBvPAKRiBVC3Nq Zd8REfjm/PkdNSgMoM/WQi+4Ll2fCt3T3d/zwqU 7jcub1E5pFrRfboWWegqe+VqZpmGNrk2VtGo0Au 5pvvEiNyMjBkvfknIMfFDFrHY8CFGpVIvN2d/ JV0JqrrVFc/zSnIq6jwzDcCM+cOgwKRpIEZa1Gak 4aFl/VLkCZGhemLeKpLFpOWIBRCzzlDrr5hQ FEBCD+pqZAxK4sv8UNOrkPS+XL/SKjmySbZ2c g==</pre>	==>	<pre>085def4766beb5f5491fc4b174ba2077e3eb4 1b382ccbd424016d0995d8d2d21cdf5d590b 8ae346b69a4858b9e963d9bec4fc6c126c3fe5 38e05a03fdfd57562a412889798f82ade1438 d406f3c0291881542dcda9977c4447e39bf3e 474d4a032833f5908bee0b9767c2b774f777f cf0a94ee372e6f5139a45ad17dba1659e82a7b e6a66998636b93656d1a8d00bb9a6fbc488d c8c8db92f7e49c831f14316b1d8f02146a552 2f37677f255d09aabad515cff34a722aea3c33 0dc08cf9c3a0c0a46920465ad466a4e1a165fd</pre>

		52e40991a17a62de2a92c5a4e5a50510b3ce50ebaf98501440420fea6a640c4ae2cbfc50d3ab90f4be5cbfd22a39b249b67672
<p>Step 2 : Take only the content of the 2 tags <typ: paramIn> and </ typ: paramIn> from received SOAP request envelope</p>		
<pre><typ:paramIn> <typ:param1>..</typ:param1> <typ:paramY> <typ:paramY_1>...</typ:paramY_1> <typ:paramY_2>...</typ:paramY_2> ... <typ:paramY_n>...</typ:paramY_n> </typ:paramY> ... <typ:paramN>xx</typ:paramN> </typ:paramIn></pre>	<p>==></p>	<pre><typ:param1>..</typ:param1> <typ:paramY> <typ:paramY_1>...</typ:paramY_1> <typ:paramY_2>...</typ:paramY_2> ... <typ:paramY_n>...</typ:paramY_n> </typ:paramY> ... <typ:paramN>xx</typ:paramN></pre>
<p>Step3: Remove all “space”, “return line”, “blank line” and “tabulation”.</p>		
<pre><typ:param1>..</typ:param1> <typ:paramY> <typ:paramY_1>...</typ:paramY_1> <typ:paramY_2>...</typ:paramY_2> ... <typ:paramY_n>...</typ:paramY_n> </typ:paramY> ... <typ:paramN>xx</typ:paramN></pre>	<p>==></p>	<pre><typ:param1>..</typ:param1> <typ:paramY> <typ:paramY_1> ... </typ:paramY_1> <typ:paramY_2> ... </typ:paramY_2> ... <typ:paramY_n> ... </typ:paramY_n> </typ:paramY> ...<typ:paramN> .. </typ:paramN></pre>
<p>STEP 4: Afterwards, a byte-representation of received request is generated via UTF-8 encoding</p>		
<p>STEP 5: Proceed to verify, the received signature via SHA2withRSA verification algorithm with input request converted in byte format.</p> <ul style="list-style-type: none"> - This verification must be done using “Sender” (SEP or Biller) public signing certificate. - The result of this algorithm is Boolean and true value means that entered signature is verified! 		
<p><i>Sha256withRSA_Verify(UTF8 byte representation of input message + decodedBase64Signature)</i></p>	<p>==></p>	<p><i>Boolean value (true : verified / false unmatched signature)</i></p>

5.3 Generation of signature for response message

Once the output response is prepared by the “Receiver” and ready to be sent back to “Sender” after processing the request, the receiver must sign the output response, and must necessary follow the below generation procedure:

<p>Step 1 : Take only the content of the 2 tags <typ: paramOut> and </ typ: paramOut> from SOAP response envelope</p>		
<pre><typ:paramOut> <typ:param1>..</typ:param1> ... <typ:paramN>xx</typ:paramN> </typ:paramOut></pre>	<p>==></p>	<pre><typ:param1>..</typ:param1> ... <typ:paramN>xx</typ:paramN></pre>
<p>Step2: Remove all “space”, “return line”, “blank line” and “tabulation”.</p>		
<pre><typ:param1>..</typ:param1> ... <typ:paramN>xx</typ:paramN></pre>	<p>==></p>	<pre><typ:param1>..</typ:param1> ... <typ:paramN> xx</typ:paramN></pre>
<p>STEP 3: Afterwards, a byte-representation of output response is generated via UTF-8 encoding</p>		
<p>STEP 4: A digital signature is then generated using receivers (SEP or Biller) signing private key for the just created byte-representation of output response via SHA2withRSA hashing algorithm.</p>		
<p><i>UTF8 byte representation</i></p>	<p>==></p>	<pre>a9a56a20edc0c033bfa2f8a86f6697441d1f67 1fbae8b1c91482fe157706e70ab3876f49bd7 802762a1a7b38c5840fbae8e9a01bf243dc87 55117ec17bcd84c904beff9c1d635008d37d3 0b187de31ec9cd6fce92993d1f2713f6f2a457 0a1bac92b1327eabd3afadc798c70a613b969 71277d06da9736405d86dec8a91c03ade4615 74eafe70723abae89ffd826c02c1efdf0d9e98e 723f3dd4ee5835d2a77f146be14d534d3bd33 7623d72dd53e94ff78ed8ee1260f64e833063 7869459695abc059c94149954aa0ff5258c31 2e54fe47fbc902b1a6e202d3753dd3ef9f4ebb c3f2ef0c363363e0dcc8541cd1958bc6e15d5 62e70888f925cf769051ada708</pre>
<p>STEP 5: Finally, the digital signature is generated through encoding the hash result to Base64String.</p>		
<pre>a9a56a20edc0c033bfa2f8a86f6697441d1f671fbae8b 1c91482fe157706e70ab3876f49bd7802762a1a7b38c 5840fbae8e9a01bf243dc8755117ec17bcd84c904beff</pre>	<p>==></p>	<pre>qaVqIO3AwDO/oviob2aXRB0fZx+66LHJF IL+FXcG5wqzh29JvXgCdioaezjFhA+66O mgG/JD3IdV</pre>

Syrian Electronic Payment Co.

Joint-Stock Company, Nominal Capital SYP 1 billion

Registered under CR#17581

الشركة السورية للمدفوعات الإلكترونية

شركة مساهمة مغلقة رأس مالها الاسمي: مليار ليرة سورية

تملك الدولة ممثلة بالخزينة العامة أسهماها كافة
مسجلة في السجل التجاري تحت الرقم 17581

9c1d635008d37d30b187de31ec9cd6fce92993d1f271 3f6f2a4570a1bac92b1327eabd3afadc798c70a613b96 971277d06da9736405d86dec8a91c03ade461574eafe 70723abae89ffd826c02c1efdf0d9e98e723f3dd4ee58 35d2a77f146be14d534d3bd337623d72dd53e94ff78e d8ee1260f64e8330637869459695abc059c94149954 aa0ff5258c312e54fe47fbc902b1a6e202d3753dd3ef9 f4ebbc3f2ef0c363363e0dcc8541cd1958bc6e15d562e 70888f925cf769051ada708	EX7Be82EyQS+/5wdY1AI030wsYfeMeyc1 vzpKZPR8nE/bypFcKG6ySsTJ+q9OvrceYx wphO5aXEn fQbalzZAXYbeyKkcA63kYVdOr+cHI6uuif /YJsAsHv3w2emOcj891O5YNdKnfXrR4U1 TTTvTN2I9 ct1T6U/3jtjuEmD2ToMwY3hpRZaVq8BZy UFJlUqg/1JYwxLIT+R/vJArGm4gLdT3T 759Ou8Py7w w2M2Pg3MhUHNGVi8bhXVYucLiPklz3aQ Ua2nCA==
--	---

5.4 Verify the signature of the response message

Once the output response is received by the “Sender”, this one must verify the signature of “Receiver” who sent the response before extracting this response and “Sender” must necessary follow the below verification procedure:

STEP 1: The received Base64String signature of receiver must be decoded to byte array format.		
qaVqIO3AwDO/oviob2aXRB0fZx+66LHJFIL+FXc G5wqzh29JvXgCdioaezjFhA+66OmgG/JD3IdV EX7Be82EyQS+/5wdY1AI030wsYfeMeyc1vzpKZP R8nE/bypFcKG6ySsTJ+q9OvrceYxwphO5aXEn fQbalzZAXYbeyKkcA63kYVdOr+cHI6uuif/YJsAs Hv3w2emOcj891O5YNdKnfXrR4U1TTTvTN2I9 ct1T6U/3jtjuEmD2ToMwY3hpRZaVq8BZyUFJlUq g/1JYwxLIT+R/vJArGm4gLdT3T759Ou8Py7w w2M2Pg3MhUHNGVi8bhXVYucLiPklz3aQUa2nC A==	==>	a9a56a20edc0c033bfa2f8a86f6697441d1f67 1fbae8b1c91482fe157706e70ab3876f49bd7 802762a1a7b38c5840fbae8e9a01bf243dc87 55117ec17bcd84c904beff9c1d635008d37d3 0b187de31ec9cd6fce92993d1f2713f6f2a457 0a1bac92b1327eabd3afadc798c70a613b969 71277d06da9736405d86dec8a91c03ade4615 74eafe70723abae89ffd826c02c1efdf0d9e98e 723f3dd4ee5835d2a77f146be14d534d3bd33 7623d72dd53e94ff78ed8ee1260f64e833063 7869459695abc059c94149954aa0ff5258c31 2e54fe47fbc902b1a6e202d3753dd3ef9f4ebb c3f2ef0c363363e0dcc8541cd1958bc6e15d5 62e70888f925cf769051ada708
Step 2 : Take only the content of the 2 tags <typ: paramOut> and </ typ: paramOut> from SOAP response envelope		
<typ:paramOut> <typ:param1>..</typ:param1> ... <typ:paramN>xx</typ:paramN> </typ:paramOut>	==>	<typ:param1>..</typ:param1> ... <typ:paramN>xx</typ:paramN>

Step3: Remove all “space”, “return line”, “blank line” and “tabulation”.

<code><typ:param1>..<</typ:param1></code> ... <code><typ:paramN>xx</typ:paramN></code>	\implies	<code><typ:param1>..<</typ:param1> ...</code> <code><typ:paramN> xx</typ:paramN></code>
---	------------	---

STEP 4: Afterwards, a byte-representation of output response is generated via UTF-8 encoding

STEP 5: Proceed to verify the received signature via SHA2withRSA verification algorithm with output response converted in byte format.

- This verification must be done using “Receiver” (SEP or Biller) public signing certificate.
- The result of this algorithm is Boolean and true value means that entered signature is verified!

<i>Sha256withRSA_Verify(UTF8 byte representation of output response + decodedBase64Signature)</i>	\implies	<i>Boolean value (true : verified / false unmatched signature)</i>
---	------------	--

6 Biller Interface Specification

6.1 Biller Interface Based On Web Service

6.1.1 Customer Authentication

- **Webservice Description:**

This webservice will be published by SEP Gateway to be invoked by any biller who is already participant of SEP Gateway platform having its password to acquire a token key that it will use in the future whenever it needs to communicate with SEP Gateway.

- **Webservice Name:**

biller_customer_authentication

- **Input Parameters:**

Field Name	Description	Type	Optional
billerCode	The biller code defined as defined in SEP Gateway	String	No
password	The biller password given by SEP Gateway once registered	String	No

- **Output Parameters:**

Field Name	Description	Type	Optional
errorCode	Represents the code of the occurred error	String	No
errorDescription	Represents the description of the occurred error	String	No
tokenKey	The generated token value after calculation	String	No

expiryDate	The date when the generated token will be expired Format : <u>YYYYMMDDHHMISS</u>	Date & Time	No
-------------------	---	-------------	----

Note: When «errorCode» equal to «000» it means no error, and the error description indicates that the process is successful.

6.1.2 Post-Paid Electronic Bill Presentment & Payment

6.1.2.1 Bill Upload

6.1.2.1.1 Bill Upload Push Mode

- **Webservice Description:**

The Bill Upload process permits the efficient transfer of bill data from Biller billing applications to SEP Gateway, and allows the biller to send a bulk of bills in the same request.

This webservice will be published by SEP Gateway to be invoked by the biller to upload multiple bills. The biller can perform multiple calls of this service to upload the needed bills in SEP Gateway.

- **Webservice Name:**

biller_bill_upload_push_mode

- **Input Parameters:**

Field Name	Description	Type	Optional				
billerCode	Represents the biller code	String	No				
tokenKey	The exchanged token during authentication phase	String	No				
uploadMode	Defines if biller will push "postpaid" bills or "prepaid" bills (vouchers) "P" for POSTPAID "R" for PREPAID	Char	No				
billingsRec	Field Name	Description	Type	Optional		Array Container	No
				POSTPAID	PREPAID		
	billingNo	Represents the billing number in the customer profile on a specific service	String	No	Yes		
	billNo	Represents the bill number for a particular billing number	String	Yes	Yes		
	billStatus	Represents the status of the uploaded bill	Bill Status Enum	No	Yes		

Syrian Electronic Payment Co.

Joint-Stock Company, Nominal Capital SYP 1 billion

Registered under CR#17581

الشركة السورية للمدفوعات الإلكترونية

شركة مساهمة مغلقة رأسمالها الاسمي: مليار ليرة سورية

تملك الدولة ممثلة بالخزينة العامة أسهمها كافة

مسجلة في السجل التجاري تحت الرقم 17581

dueAmount	Represents the due amount of the uploaded bill	Decimal	No	Yes
issueDate	Represents the date when the bill was issued by the biller <u>Format :</u> <u>YYYYMMDDHHMISS</u>	Date & Time	No	Yes
openDate	Represents the date when the bill is available for payment over SEP Gateway Network <u>Format :</u> <u>YYYYMMDDHHMISS</u>	Date & Time	Yes	Yes
dueDate	Represents the date when the bill is be ingrequired to pay <u>Format :</u> <u>YYYYMMDDHHMISS</u>	Date & Time	No	Yes
expiryDate	Represents the date when the bill is nolonger available for payment over SEP Gateway network but still Inquirable In case of prepaid voucher, represents the date after it the prepaid voucher will be invalide and cannot be saled or used <u>Format :</u> <u>YYYYMMDDHHMISS</u>	Date & Time	Yes	No
closeDate	Represents the date when the bill is no longer payable or inquirable over SEP Gateway Network <u>Format :</u> <u>YYYYMMDDHHMISS</u>	Date & Time	Yes	Yes
serviceType	Represents the service type of the Bill or PrePaid Voucher If the payment mode of service type is "POSTPAID" the billingsRec concerns only the bills	Service Type Enum	No	No

Syrian Electronic Payment Co.

Joint-Stock Company, Nominal Capital SYP 1 billion

Registered under CR#17581

الشركة السورية للمدفوعات الإلكترونية

شركة مساهمة مغلقة رأسمالها الاسمي: مليار ليرة سورية

تملك الدولة ممثلة بالخزينة العامة أسهمها كافة
مسجلة في السجل التجاري تحت الرقم 17581

		Else if the payment mode is "PREPAID" the billingsRec concerns only the prepaid bills (prepaid vouchers)				
billType		Represents the type of the bill	Bill Type Enum	No	Yes	
allowPart		Represents the modality to pay the bill partially or exact "Y" : Allow Partial Payment "N" : Partial Payment not allowed	Char	Yes	Yes	
lowestToPay		Represents the lowest possible value to pay	Decimal	Yes	Yes	
highestToPay		Represents the highest possible value to pay	Decimal	Yes	Yes	
allowOver		Represents the modality to pay the bill with over payment or exact "Y" : Allow Over Payment "N" : Over Payment not allowed	Char	Yes	Yes	
highestToOverPay		Represents the highest possible value to pay for over payment	Decimal	Yes	Yes	
denomination		Represents the pre-paid denomination ID related to this service type if it exists	Pre-paid Category Enum	Yes	No	
serialNumber		Represents the unique serial number of the prepaid voucher the may identify it		Yes	No	
validationCode		Represents the secret pin code of prepaid vouchers communicated by the biller to SEP Gateway with RSA encryption using public signing certification of SEP (same used to verify signature of SEP)	String	Yes	No	
message		Information message should be sent to the customer during the bill presentment	String	Yes	Yes	

- Output Parameters:

Tel: +963 11 3937050 – Fax: +963 11 3937028- website: www.sep.com.sy - Email: info@sep.com.sy

Field Name	Description	Type
errorCode	Represents the code of the occurred error	String
errorDescription	Represents the description of the occurred error	String

Note: When «errorCode» equal to «000» it means no error, and the error description indicates that the process is successful.

6.1.2.1.2 Bill Upload Pull Mode

- Webservice Description:

EBPP Gateway may 'pull' bill data from the Biller site by sending the Bill Pull message. The Biller billing application must respond with the bill data of the requested bills.

Biller upload bills based on SEP Gateway request which includes certain input parameters. This table illustrates how this webservice that is published by the Biller should be invoked by SEP Gateway. This table illustrates how this webservice that is published by the Biller should be invoked by SEP Gateway.

Upload Mode	Bill No	Billing No	Service Type	Action
Real Time	Null	Null	Null	The biller should upload all bills in real time as a response of the webservice
Real Time	Null	Null	Value	The biller should upload all bills related to a specific service type in real time as a response of the webservice
Real Time	Null	Value	Value	The biller should upload all bills related to a specific billing number under a specific service type in real time as a response of the webservice
Real Time	Value	Value	Value	The biller should upload the bill related to a specific billing number under a specific service type in real time as a response of the webservice
Scheduled Time	Null	Null	Null	The biller should upload, using push mode at scheduled time as agreed between SEP Gateway and the biller, all bills.

Syrian Electronic Payment Co.

Joint-Stock Company, Nominal Capital SYP 1 billion

Registered under CR#17581

الشركة السورية للمدفوعات الإلكترونية

شركة مساهمة مغفلة رأسمالها الاسمي: مليار ليرة سورية

تملك الدولة ممثلة بالخبزينة العامة أسهمها كافة
مسجلة في السجل التجاري تحت الرقم 17581

Scheduled Time	Null	Null	Value	The biller should upload, using push mode at scheduled time as agreed between SEP Gateway and the biller, all bills related to a specific service type.
Scheduled Time	Null	Value	Value	The biller should upload, using push mode at scheduled time as agreed between SEP Gateway and the biller, all bills related to a specific billing number under a specific service type.
Scheduled Time	Value	Value	Value	The biller should upload, using push mode at scheduled time as agreed between SEP Gateway and the biller, the bill related to a specific billing number under a specific service type.

- **Webservice Name:**

biller_bill_upload_pull_mode.

- **Input Parameters:**

Field Name	Description	Type	Optional			
billerCode	Represents the biller code	String	No			
uploadMode	Flag to indicate if this upload request must be processed on real time or just scheduled by biller for earlier. "R" = "Real Time upload" "S" = "Scheduled upload"	Char	No			
billingsRec	Name	Description	Type	Container	Yes	
	billingNo	Represents the billing number in the customer profile on a specific service	String			No
	billNo	Represents the bill number for a particular billing number	String			Yes
	serviceType	Represents the service type of the Bill	Service Type Enum			No

- **Output Parameters:**

Field Name	Description	Type	Optional				
			RealTime	Scheduled			
errorCode	Represents the code of the occurred error	String	No	No			
errorDescription	Represents the description of the occurred error	String	No	No			
billingsRec	Name	Description	Type	Optional	Array Container	No	Yes
	billingNo	Represents the billing number in the customer	String	No			

Syrian Electronic Payment Co.

Joint-Stock Company, Nominal Capital SYP 1 billion

Registered under CR#17581

الشركة السورية للمدفوعات الإلكترونية

شركة مساهمة مغلقة رأسمالها الاسمي: مليار ليرة سورية

تملك الدولة ممثلة بالخبزينة العامة أسهمها كافة

مسجلة في السجل التجاري تحت الرقم 17581

		profile on a specific service			(if successful)
billNo		Represents the bill number for a particular billing number	String	Yes	
billStatus		Represents the status of the uploaded bill	Bill Status Enum	No	
dueAmount		Represents the due amount of the uploaded bill	Decimal	No	
issueDate		Represents the date when the bill was issued by the biller Format : <u>YYYYMMDDHHMISS</u>	Date & Time	No	
openDate		Represents the date when the bill is available for payment over SEP Gateway Network Format : <u>YYYYMMDDHHMISS</u>	Date & Time	Yes	
dueDate		Represents the date when the bill is being required to pay Format : <u>YYYYMMDDHHMISS</u>	Date & Time	No	
expiryDate		Represents the date when the bill is no longer available for payment over SEP Gateway network but still Inquirable Format : <u>YYYYMMDDHHMISS</u>	Date & Time	Yes	
closeDate		Represents the date when the bill is no longer payable or inquirable over SEP Gateway Network Format : <u>YYYYMMDDHHMISS</u>	Date & Time	Yes	
serviceType		Represents the service type of the Bill	Service Type Enum	No	

billType	Represents the type of the bill	Bill Type Enum	No			
allowPart	Represents the modality to pay the bill partially or exact "Y" : Allow Partial Payment "N" : Partial Payment not allowed	Char	Yes			
lowestToPay	Represents the lowest possible value to pay	Decimal	Yes			
highestToPay	Represents the highest possible value to pay	Decimal	Yes			
allowOver	Represents the modality to pay the bill with over payment or exact "Y" : Allow Over Payment "N" : Over Payment not allowed	Char	Yes			
highestToOverPay	Represents the highest possible value to pay for over payment	Decimal	Yes			
message	Information message should be sent to the customer during the bill presentment	String	Yes			

Note: When «errorCode» equal to «000» it means no error, and the error description indicates that the process is successful.

6.1.2.2 Bill Presentment

- Webservice Description:

This webservice will be published by biller to be invoked by the SEP Gateway once the inquiry request initiated by the bank when the end customer needs to inquiry his bills.

Bill presentment request from SEP Gateway to billers should include certain input parameters.

This table illustrates how this webservice should be invoked by the SEP Gateway.

The biller should response in Thin or Thick model depending on the Inquiry mode defined in the input parameter of this web service.

This web service allows the SEP Gateway sending one presentment request or multiple presentment requests for different bills under same billing number (Thick Biller) or different billing numbers under same biller (Thin Biller) And different bills under different billing numbers under same biller.

Bill No	Billing No	Service Type	Date From-To	IncPaid Bills	Action
Null	Value	Value	Null	Y/N	SEP Gateway inquiry the biller to provide all bills data (Thick billers) or summary data (Thin billers) under a billing No/Service Type including or not paid bills depending on the flag of IncPaidBills
Value	Value	Value	Null	Y/N	SEP Gateway inquiry the biller to provide the bill data under a billing No/Service Type in case IncPaidBills Flag to include paid bills. In case IncPaidBills Flag not to include paid bills then the bill data will be returned if it is not paid. This parameter is not applicable to the Thin billers.
Null	Value	Value	Value	Y/N	SEP inquiry the biller to provide all bills data (Thick billers) or summary data (Thin billers) under a billing No/Service Type including or not paid bills depending on the flag of IncPaidBills with the due date between the parameters (Date From – To)

- **Webservice Name:**

biller_bill_presentment

- **Input Parameters:**

Field Name	Description	Type	Optional
billerCode	Represents the biller code	String	No

Syrian Electronic Payment Co.

Joint-Stock Company, Nominal Capital SYP 1 billion

Registered under CR#17581

الشركة السورية للمدفوعات الإلكترونية

شركة مساهمة مغلقة رأسمالها الاسمي: مليار ليرة سورية

تملك الدولة ممثلة بالخبزينة العامة أسهمها كافة
مسجلة في السجل التجاري تحت الرقم 17581

bankCode	Represents the bank code who initiate the bill presentment inquiry				String	No
inquiryMode	"O" : Inquiry Only "P" : Inquiry for Payment				Char	No
billingsRec	Name	Description	Type	Optional	Array Container	No
	billingNo	Represents the billing number in the customer profile on a specific service	String	No		
	billNo	Represents the bill number for a particular billing number	String	Yes		
	serviceType	Represents the service type of the Bill	Service Type Enum	No		
	dateFrom	Represents the value that will be entered into inquiry process to load all bills after this date considering the Bill Due date Format : YYYYMMDDHHMISS	Date & Time	Yes		
	dateTo	Represents the value that will be entered into inquiry process to load all bills before this date considering the Bill Due date Format : YYYYMMDDHHMISS	Date & Time	Yes		
incPaidBills	Flag to include or not the paid bills "N" Only unpaid bills "Y" Include fully paid bills also		Char	No		

- Output Parameters:

Field Name	Description	Type	Optional			
errorCode	Represents the code of the occurred error	String	No			
errorDescription	Represents the description of the occurred error	String	No			
billingsRec	Name	Description	Type	Optional	Array Container	No
	billingNo	Represents the billing number in the customer profile on a specific service	String	No		
	billNo	Represents the bill number for a particular billing number	String	No		
	billStatus	Represents the status of the uploaded bill	Bill Status Enum	No		
	dueAmount	Represents the due amount of the uploaded bill	Decimal	No		

Syrian Electronic Payment Co.

Joint-Stock Company, Nominal Capital SYP 1 billion

Registered under CR#17581

الشركة السورية للمدفوعات الإلكترونية

شركة مساهمة مغلقة رأسمالها الاسمي: مليار ليرة سورية

تملك الدولة ممثلة بالخبزينة العامة أسهمها كافة
مسجلة في السجل التجاري تحت الرقم 17581

issueDate	Represents the date when the bill was issued by the biller <u>Format :</u> <u>YYYYMMDDHHMISS</u>	Date & Time	No		
openDate	Represents the date when the bill is available for payment over SEP Gateway Network <u>Format :</u> <u>YYYYMMDDHHMISS</u>	Date & Time	Yes		
dueDate	Represents the date when the bill is being required to pay <u>Format :</u> <u>YYYYMMDDHHMISS</u>	Date & Time	No		
expiryDate	Represents the date when the bill is no longer available for payment over SEP Gateway network but still Inquirable <u>Format :</u> <u>YYYYMMDDHHMISS</u>	Date & Time	Yes		
closeDate	Represents the date when the bill is no longer payable or inquirable over SEP Gateway Network <u>Format :</u> <u>YYYYMMDDHHMISS</u>	Date & Time	Yes		
serviceType	Represents the service type of the Bill	Service Type Enum	No		
billType	Represents the type of the bill	Bill Type Enum	No		
allowPart	Represents the modality to pay the bill partially or exact	Boolean	Yes		
lowestToPay	Represents the lowest possible value to pay	Decimal	Yes		
highestToPay	Represents the highest possible value to pay	Decimal	Yes		
allowOver	Represents the modality to pay the bill with over payment or exact	Boolean	Yes		
highestToOverPay	Represents the highest	Decimal	Yes		

		possible value to pay for over payment				
	message	Information message should be sent to the customer during the bill presentment	String	Yes		

Note: When «errorCode» equal to «000» it means no error, and the error description indicates that the process is successful.

6.1.2.3 Bill Payment Request

The Bill Payment process permits Banks to create new payment records in SEP Gateway. The process is intended to involve a set of validations on the received payment, if the validation process is successful, SEP Gateway will generate a payment transaction number 'EBPPSTrx' and return it in the response message to the bank.

- Webservice Description:

This webservice will be published by Biller to be invoked by the SEP Gateway once the payment request is initiated by the Bank when the end customer needs to pay his bills.

This web service allows the SEP Gateway sending one payment request or multiple payment requests for different bills under same billing number (Thick Biller) or different billing numbers under same biller (Thin Biller) and different bills under different billing numbers under same biller. If the SEP Gateway sends multiple payment requests then the Biller will process them as a group to validate or reject all of them.

Bill No	Billing No	Service Type	Action
Null	Value	Value	SEP Gateway sends the payment request for the total due of the entered billing No/Service Type (Thin billers).
Value	Value	Value	SEP Gateway sends the payment request for the bill amount of the entered Bill No/billing No/Service Type (Thick billers).

- Webservice Name:

biller_bill_payment

- Input Parameters:

Syrian Electronic Payment Co.

Joint-Stock Company, Nominal Capital SYP 1 billion

Registered under CR#17581

الشركة السورية للمدفوعات الإلكترونية

شركة مساهمة مغلقة رأسمالها الاسمي: مليار ليرة سورية

تملك الدولة ممثلة بالخرزينة العامة أسهمها كافة
مسجلة في السجل التجاري تحت الرقم 17581

Field Name	Description	Type	Optional			
billerCode	Represents the biller code	String	No			
bankCode	Represents the bank code who initiate the bill payment request	String	No			
transRec	Name	Description	Type	Optional	Array Container	No
	billingNo	Represents the billing number in the customer profile on a specific service	String	No		
	billNo	Represents the bill number for a particular billing number	String	Yes		
	serviceType	Represents the service type of the Bill	Service Type Enum	No		
	bankTrxId	Represents the bank transaction number that generated when collect the payment	String	No		
	EBPPSTrx	Represents the EBPP Gateway transaction number that used as a reference number between all banks, billers and EBPP Gateway	String	No		
	stmtDate	Represents the date when the payment will be settled if approved Format : <u>YYYYMMDD</u>	Date	No		
	pmtStatus	Represents the status of the collected payment	Payment Status Enum	No		
	dueAmt	Represents the amount of the bill	Decimal	No		
	paidAmt	Represents the collected payment amount	Decimal	No		
	processDate	Represents when the payment processed at bank side Format : <u>YYYYMMDDHHMISS</u>	Date & Time	No		
	accessChannel	Represents the payment channel type that used to collect the payment	Access Channel Enum	No		
paymentMethod	Represents the method of the payment via bank access channel	Payment Method Enum	No			

	paymentType	The type of the payment being processed in terms of collection period like yearly fees, monthly fees, one-shot etc...	Payment Type enum	No		
	currency	The type of the currency that used in payment process	Currency Type Enum	No		

- **Output Parameter:**

Field Name	Description	Type
errorCode	Represents the code of the occurred error	String
errorDescription	Represents the description of the occurred error	String

Note: When «errorCode» equal to «000» it means no error, and the error description indicates that the process is successful.

6.1.2.4 Payment Notification

Payment Notification messages are used to notify the biller that the payment request that was received from SEP Gateway and validated by himself was confirmed by the bank.

- **Webservice Description:**

This webservice will be published by Biller to be invoked by the SEP Gateway once the payment request, that was initiated by the Bank when the end customer has paid his bills, is confirmed by the bank.

This web service allows the SEP Gateway sending one payment notification or multiple payment notifications for different bills under same billing number (Thick Biller) or different billing numbers under same biller (Thin Biller). And different bills under different billing numbers under same biller.

Bill No	Billing No	Service Type	Action
Null	Value	Value	SEP Gateway sends the payment notification for the total due of the entered billing No/Service Type (Thin billers).

Syrian Electronic Payment Co.

Joint-Stock Company, Nominal Capital SYP 1 billion

Registered under CR#17581

الشركة السورية للمدفوعات الإلكترونية

شركة مساهمة مغلقة رأسمالها الاسمي: مليار ليرة سورية

تملك الدولة ممثلة بالخبزينة العامة أسهمها كافة

مسجلة في السجل التجاري تحت الرقم 17581

Value	Value	Value	SEP Gateway sends the payment notification for the bill amount of the entered Bill No/billing No/Service Type (Thick billers).
-------	-------	-------	--

- **Webservice Name:**

biller_bill_payment_notif

- **Input Parameters:**

Field Name	Description	Type	Optional			
billerCode	Represents the biller code	String	No			
bankCode	Represents the bank code who confirmed the bill payment	String	No			
transRec	Name	Description	Type	Optional	Array Container	No
	billingNo	Represents the billing number in the customer profile on a specific service	String	No		
	billNo	Represents the bill number for a particular billing number	String	Yes		
	serviceType	Represents the service type of the Bill	Service Type Enum	No		
	bankTrxId	Represents the bank transaction number that generated when collect the payment	String	No		
	EBPPSTrx	Represents the EBPP Gateway transaction number that used as a reference number between all banks, billers and EBPP Gateway	String	No		
	stmtDate	Represents when the payment will be Settled Format : <u>YYYYMMDD</u>	Date	No		
	pmtStatus	Represents the status of the collected payment	Payment Status Enum	No		
	dueAmt	Represents the amount of the bill	Decimal	No		
	paidAmt	Represents the collected payment amount	Decimal	No		
processDate	Represents when the payment processed at bank side Format : <u>YYYYMMDDHHMISS</u>	Date & Time	No			

	accessChannel	Represents the payment channel type that used to collect the payment	Access Channel Enum	No		
	paymentMethod	Represents the method of the payment via bank access channel	Payment Method Enum	No		
	paymentType	The type of the payment being processed in terms of collection period like yearly fees, monthly fees, one-shot etc...	Payment Type enum	No		
	currency	The type of the currency that used in payment process	Currency Type Enum	No		

- **Output Parameters:**

Field Name	Description	Type	Optional
errorCode	Represents the code of the occurred error	String	No
errorDescription	Represents the description of the occurred error	String	No

Note: When Error Code equal to zero it means no error, and the error description indicates that the process is successful.

6.1.3 Pre-Paid Validation

6.1.3.1 Pre-paid Inquiry

- **Webservice Description:**

This webservice will be published by the Biller to be invoked by SEP Gateway. This webservice will be used so that SEP Gateway sends pre-paid payment inquiry received from the Bank when the end customer chooses the pre-paid service and the denomination that he wants to pay. The output of this webservice will be sent to the Bank to display to the end customer to confirm or reject the payment.

- **Webservice Name:**

biller_prepaid_inquiry

- **Input Parameters:**

Field Name	Description	Type	Optional			
billerCode	Represents the biller code	String	No			
bankCode	Represents the bank code who initiate the prepaid inquiry	String	No			
prepaidRec	Name	Description	Type	Optional	Container	No

	billingNo	Represents the billing number in the customer profile on a specific service	String	No		
	serviceType	Represents service type of the prepaid bill	Service Type Enum	No		
	denomination	Represents the pre-paid denomination related to this service type if it exists	Pre-paid Category Enum	<u>Condition</u> ⁽¹⁾		
	dueAmount	Represent a flexible due Amount in case the denomination is not applicable for this service type	Decimal	<u>Condition</u> ⁽²⁾		

Notes:

⁽¹⁾ When a list of denominations is related to the service type, the field denomination must be present and represent the one of denomination ID configured on SEP Gateway and shared/agreed with billers and Banks. In this case, the dueAmount must be empty.

⁽²⁾ When no denomination is related to the indicated service type like “e-topup service with flexible amount”, the dueAmount field must be indicated and denomination field must be empty.

- **Output Parameters:**

Field Name	Description	Type	Optional			
errorCode	Represents the code of the occurred error	String	No			
errorDescription	Represents the description of the occurred error	String	No			
prepaidRec	Name	Description	Type	Optional	Container	No
	dueAmount	Represents the amount corresponding to the chosen denomination if applicable for this service type or entered amount	Decimal	No		
	message	Information message should be sent to the customer during prepaid inquiry	String	Yes		

Note: When «errorCode» equal to «000» it means no error, and the error description indicates that the process is successful.

6.1.3.2 Pre-paid Payment

- **Webservice Description:**

This webservice will be published by the Biller to be invoked by SEP Gateway. This webservice will be used so that the SEP Gateway sends pre-paid payment request to the Biller, this payment

request that is initiated by the Bank when the end customer confirms the payment of the pre-paid denomination.

- **Webservice Name:**

biller_prepaid_payment

- **Input Parameters:**

Field Name	Description	Type	Optional			
billerCode	Represents the biller code	String	No			
bankCode	Represents the bank code who initiate the prepaid payment request	String	No			
transRec	Name	Description	Type	Optional	Container	No
	billingNo	Represents the billing number in the customer profile on a specific service	String	No		
	serviceType	Represents the service type of the prepaid bill	Service Type Enum	No		
	denomination	Represents the pre-paid denomination related to this service type if it exists	Pre-paid Category Enum	<u>Condition</u> (3)		
	bankTrxId	Represents the bank transaction number that generated when collect the payment	String	No		
	EBPPSTrx	Represents the EBPP Gateway transaction number that used as a reference number between all banks, billers and EBPP Gateway	String	No		
	stmtDate	Represents the date when the payment will be settled if approved <u>Format : YYYYMMDD</u>	Date	No		
	pmtStatus	Represents the status of the collected payment	Payment Status Enum	No		
	dueAmt	Represents the amount of the prepaid bill	Decimal	No		
	paidAmt	Represents the collected payment amount	Decimal	No		
processDate	Represents when the payment processed at bank side <u>Format : YYYYMMDDHHMISS</u>	Date & Time	No			

	accessChannel	Represents the payment channel type that used to collect the payment	Access Channel Enum	No		
	paymentMethod	Represents the method of the payment via bank access channel	Payment Method Enum	No		
	paymentType	The type of the payment being processed in terms of collection period like yearly fees, monthly fees, one-shot etc...	Payment Type num	No		
	currency	The type of the currency that used in payment process	Currency Type Enum	No		

Note:

⁽³⁾ When a list of denominations is related to the service type, the field denomination must be present and represent the one of denomination ID configured on SEP Gateway and shared/agreed with billers and Banks.

- **Output Parameters:**

Field Name	Description	Type	Optional
Error Code	Represents the code of the occurred error	String	No
Error Description	Represents the description of the occurred error	String	No

Note: When Error Code equal to zero it means no error, and the error description indicates that the process is successful.

6.1.3.3 Pre-Paid Payment Notification

- **Webservice Description:**

This webservice will be published by the Biller to be invoked by SEP Gateway. This webservice will be used so that SEP Gateway sends pre-paid payment notification to the biller once the pre-paid payment confirmation is received from the Bank.

- **Webservice Name:**

biller_prepaid_payment_notif

- **Input Parameters:**

Field Name	Description	Type	Optional
billerCode	Represents the biller code	String	No

Syrian Electronic Payment Co.

Joint-Stock Company, Nominal Capital SYP 1 billion

Registered under CR#17581

الشركة السورية للمدفوعات الإلكترونية

شركة مساهمة مغلقة رأس مالها الاسمي: مليار ليرة سورية
تملك الدولة ممثلة بالخرزينة العامة أسهمها كافة
مسجلة في السجل التجاري تحت الرقم 17581

bankCode	Represents the bank code who confirm the prepaid payment				String	No
transRec	Name	Description	Type	Optional	Container	No
	billingNo	Represents the billing number in the customer profile on a specific service	String	No		
	serviceType	Represents the service type of the prepaid bill	Service Type Enum	No		
	denomination	Represents the pre-paid denomination related to this service type if it exists	Pre-paid Category Enum	<u>Condition</u> ⁽³⁾		
	bankTrxId	Represents the bank transaction number that generated when collect the payment	String	No		
	EBPPSTrx	Represents the EBPP Gateway transaction number that used as a reference number between all banks, billers and EBPP Gateway	String	No		
	stmtDate	Represents when the payment will be Settled Format : <u>YYYYMMDD</u>	Date	No		
	pmtStatus	Represents the status of the collected payment	Payment Status Enum	No		
	dueAmt	Represents the amount of the prepaid bill	Decimal	No		
	paidAmt	Represents the collected payment amount	Decimal	No		
	processDate	Represents when the payment processed at bank side Format : <u>YYYYMMDDHHMISS</u>	Date & Time	No		
	accessChannel	Represents the payment channel type that used to collect the payment	Access Channel Enum	No		
	paymentMethod	Represents the method of the payment via bank access channel	Payment Method Enum	No		
paymentType	The type of the payment being processed in terms of collection period like yearly fees, monthly fees, one-shot etc...	Payment Typenum	No			

Syrian Electronic Payment Co.

Joint-Stock Company, Nominal Capital SYP 1 billion

Registered under CR#17581

الشركة السورية للمدفوعات الإلكترونية

شركة مساهمة مغلقة رأسمالها الاسمي: مليار ليرة سورية

تملك الدولة ممثلة بالخرزينة العامة أسهمها كافة

مسجلة في السجل التجاري تحت الرقم 17581

	currency	The type of the currency that used in payment process	Currency Type Enum	No		
--	-----------------	---	--------------------	----	--	--

Note:

(3) When a list of denominations related to the service type, the field denomination must be present and represent the one of denomination ID configured on SEP Gateway and shared/agreed with biller and Banks.

- Output Parameters:

Fiel Name	Description	Type	Optional
errorCode	Represents the code of the occurred error	String	No
errorDescription	Represents the description of the occurred error	String	No

Note: When «errorCode» equal to «000» it means no error, and the error description indicates that the process is successful.