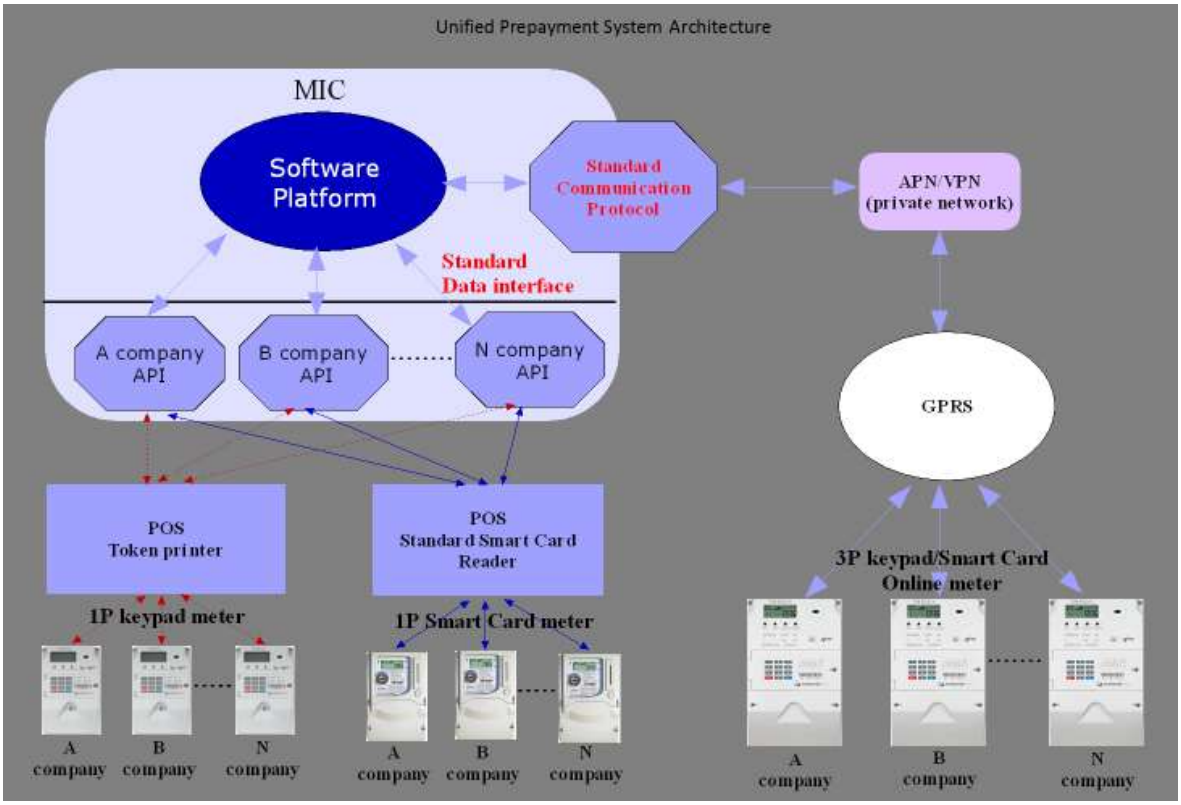
(Unified Prepayment System API)

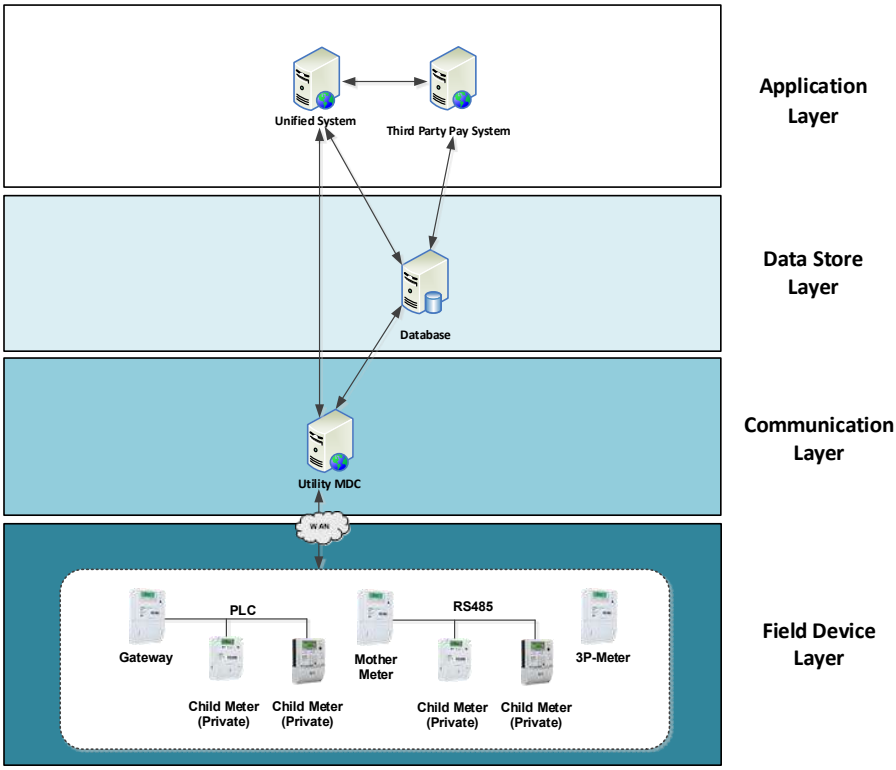
1 Multi-factories Access Platform

1.1 Access Platform Architecture Design



Each manufacturer provides its own encryption / decryption API, integrated into the system software.
When dealing withbusiness based ondifferent meters from each manufacturer, the system automaticallycallsthe correspondingmanufacturers'encryption/ decryptionAPI.
The systemdefines uniformly data exchange interfacecorresponding todifferent business, such as input and outputparameters.
When dealing withabusiness,for thekeypadmeter, the system will call the encryption/ decryptionAPI,generatesthe correspondingToken, and print; for smart cardmeter, the system will call the encryption/ decryptionAPI,write thecorresponding Tokento encrypted data area of thesmart card.
The encryption/decryption API from smart card manufacture must be able to generate Token which must contain the information of business type and length information. So that when smart card meter read the encrypted data in the card, the meter can separate the Token and identify the Token business itself.
POS read the smart cards from different manufacturesviastandardcardread devices.
MIC communicates with GPRS meters of other factories according to standard data exchangemechanism.
Note: Token in this document means encrypted data.

1.2Online Communication System Architecture



For online meter communication, all manufacturers should communicate with their online meters by WZPDCL`s Head End System (HES).

And the related documents and updated API, please refer to

Annexure- A : Unified Meter Data Collection System: Addressing System；

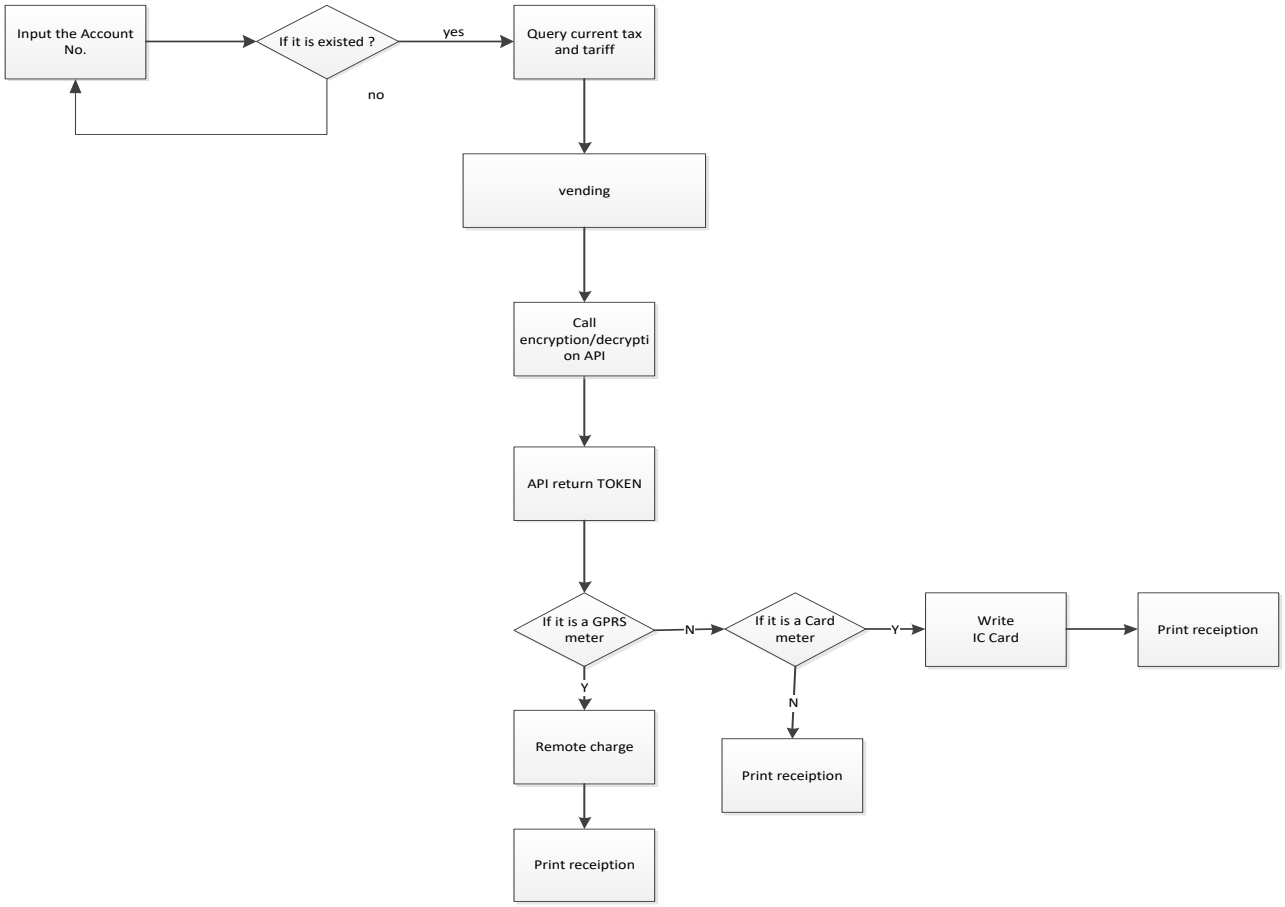
Annexure- B: Unified Meter Data Collection System: Meter Modeling；

Annexure- C: Unified Meter Data Collection System: DCU to HES；

Annexure- D: Unified System &HES Interface Specification;

2. Business Process

2.1 Vending as example:



3. Keypad Meters and Card Meter Access design

3.1 ErrorCode definition:

0 or empty: success

1: sequence number is not excepted

2: tariff Index argument error, not in scope

3: keyVersion argument not in scope

4: keyExpiredTime not in scope(0-255)

5: keyNo exceed 65535 or < 0

6: meterNo is not excepted

7: credit amount <= 0

8: unknown reason

3.2 Standard Interface for Generating A Variety of Token

3.2.1 Token Standard Interface Define

Programming language	JAVA
Functionality	Token standard interface class
Package	com.hx.ami.spi
Interface	Token

Class definition	public interface Token
	{
	Generatecredit Token
	public String getCreditToken(String meterNo, String sgc, int tarrifIndex, int keyVersion, int keyExpiredTime, long keyNo, int seqNo, double amount);
	
	Generate key change Token
	public String getKeyChangeToken (String meterNo, String sgc, int tarrifIndex, int keyVersion, int keyExpiredTime, long keyNo, String nSgc, int nTariffIndex, int nKeyVersion,int nKeyExpiredTime, long nKeyNo);
	
	Generate Management Token
	
	Generate clear balance Token
	public String getClearBalanceToken (String meterNo, String Sgc, int tarrifIndex, int keyVersion, int keyExpiredTime, long keyNo, int seqNo);
	
	Generate clear event Token
	public String getClearEventToken (String meterNo, String Sgc, int tarrifIndex, int keyVersion, int keyExpiredTime, long keyNo, int seqNo);
	
	Generate max Power Limit Mode Token
	public String getMaxPowerLimitToken(String meterNo,String Sgc, int tarrifIndex, int keyVersion, int keyExpiredTime, long keyNo, int seqNo, int activationModel, String date,int[] maxPowerLimits, int[] hours);
	
	Generate single Tariff Token
	public String getSingleTariffToken(String meterNo, String Sgc, int tarrifIndex,int keyVersion, int keyExpiredTime, long keyNo, int seqNo,String activatingDate, int activatingModel, int validDate, int rate);
	
	Generate step Tariff Token
	public String getStepTariffToken(String meterNo, String Sgc, int tarrifIndex,int keyVersion, int keyExpiredTime, long keyNo, int seqNo,String activatingDate, String validDate, int[] rates, int[] steps);
	Generate Tou Tariff Token
	public String getTOUTariffToken(String meterNo, String Sgc, int tarrifIndex,int keyVersion, int keyExpiredTime, long keyNo, int seqNo,String activatingDate, int activatingModel, int validDate, int[] rates, int[] times);

	Generate Friend Mode Token
	public String getFriendModeToken(String meterNo, String Sgc, int tarrifIndex, int keyVersion, int keyExpiredTime, long keyNo, int seqNo, int friendMode, int[] times, int[] days);
	
	Generate Public Holiday Token
	generateHolidayModeToken(String meterNo,String sgcId, int ti, int kv, int ke, int seq, int keyNo,int holidayMode, String[] days);
	
	Generate Change Meter Mode Token
	public String getChangeMeterModeToken(String meterNo, String Sgc,
	int tarrifIndex, int keyVersion, int keyExpiredTime, long keyNo, int seqNo,int mode);
	
	Generate Set Credit Amount Limit And Overdraw Amount Limit Token
	public String getSetCreditAmountLimitOrOverdrawAmountLimitToken(
	String meterNo, String Sgc, int tarrifIndex, int keyVersion, int keyExpiredTime, long keyNo, int seqNo, int amountType, int amountLimit);
	
	Generate Set Low Credit Warning Limit Token
	public String SetLowCreditWarningLimitToken(
	String meterNo, String Sgc, int tarrifIndex, int keyVersion, int keyExpiredTime, long keyNo, int seqNo, int amountType, int amountLimit);
	
	Generate LogoffReturn Token
	
	public String generateLogoffReturnToken (String meterNo, String Sgc, int tarrifIndex, int keyVersion, int keyExpiredTime, int keyNo, int seqNo);
	
	Resolve the Return Token
	public String resolveReturnToke(String meterNo, String Sgc, int tarrifIndex, int keyVersion, int keyExpiredTime, long keyNo, String Token);
	
	4) Generate **Test Token**
	public String getTestToken (int manufacturingID, int control);
	}

3.2.2 Charge Interface Definition

Functionality	To get credit Token
Application	To call the function and generate credit token when vending. seqNo : Each token has sequence number except change-key-token. seqNo will increase 1 for each token generated and change to 1 when above 200; keyNo: when seqNo change from 200 to 1,keyNo add 1. keyExpiredTime: Global system parameter used as cipher fact. System can change

<table>
<tr><td></td><td>it. Nothing to do with key expiration. Just a encryption fact.
Key version: Global encryption fact, it will be changed in case all meter need to change keys.</td></tr>
<tr><td>Function name</td><td>getCreditToken</td></tr>
<tr><td>Parameter list</td><td>
<table>
<tr><th>Parameter name</th><th>Type</th><th>Scope or length</th><th>Description</th></tr>
<tr><td>meterNo</td><td>String</td><td>String of 12 numbers</td><td>Meter number</td></tr>
<tr><td>Sgc</td><td>String</td><td>String of 6 numbers</td><td>Supply group code
System parameter.</td></tr>
<tr><td>tarrifIndex</td><td>Int</td><td>1-99</td><td>Tariff index
When custom change tariff, this value may change to any of 1-99</td></tr>
<tr><td>keyVersion</td><td>Int</td><td>1-9</td><td>Key version
Global encryption fact, it will be changed in case all meter need to change keys.</td></tr>
<tr><td>keyExpiredTime</td><td>Int</td><td>0-255</td><td>Global system parameter used as cipher fact. System can change it.
Nothing to do with key expiration. Just a encryption fact</td></tr>
<tr><td>keyNo</td><td>Long</td><td>0-65535</td><td>Key Sequence, similar to key changed times, see above method descpt.</td></tr>
<tr><td>seqNo</td><td>Int</td><td>1-200</td><td>Token Sequence</td></tr>
<tr><td>amount</td><td>Int</td><td>0-99999999(scale&unit 0.01TK)</td><td>Credit amount</td></tr>
</table>
</td></tr>
<tr><td>Return value</td><td>Return XML String
Xml format:
<?xml version="1.0" encoding="UTF-8"?>
<result>
<errorCode></errorCode>
<tokens>
<token></token>
...
<token></token>
<tokens></td></tr>
</table>

	</result> Note： errorCode see 3. Chart definition. If succeeded, need return an array of <tokens>elements.

Function definition	public String getCreditToken(String meterNo, String sgc, int tarrifIndex,
	int keyVersion, int keyExpiredTime, long keyNo, int seqNo, int amount)
	{
	…
	}

3.2.3 Key ChangeInterface Definition

<table>
<tr><td>Functionality</td><td>To get key change Token</td></tr>
<tr><td>Application</td><td>As soon as user wants to charge money , to generate the manage token and test token ,and judge if the encryption factor has been changed, if changed , to call function and generate change token.
This function is called at four case:
Meter installed and custom purchase energy at first time
Called if tariff changed
SeqNo change from 200 to 1.
Called when operator press menu to change key.

For the first time registration, parameter defines as below:
tariffIndex=0; keyVersion=0; keyExpiredTime=0; keyNo=0;</td></tr>
<tr><td>Function name</td><td>getKeyChangeToken</td></tr>
<tr><td>Parameter list</td><td>
<table>
<tr><th>Parameter name</th><th>Type</th><th>Scope or length</th><th>Description</th></tr>
<tr><td>meterNo</td><td>String</td><td>String of 12 numbers</td><td>Meterno.</td></tr>
<tr><td>Sgc</td><td>String</td><td>String of 6 numbers</td><td>Old Supply group code</td></tr>
<tr><td>tariffIndex</td><td>Int</td><td>1-99</td><td>Old Tariff index</td></tr>
<tr><td>keyVersion</td><td>Int</td><td>1-9</td><td>Old Key version</td></tr>
<tr><td>keyExpiredTime</td><td>Int</td><td>0-255</td><td>Old Key expire time</td></tr>
<tr><td>keyNo</td><td>Long</td><td>0-65535</td><td>Old Key Sequence</td></tr>
<tr><td>nSgc</td><td>Int</td><td>String of 6 numbers</td><td>Current supply grop code</td></tr>
<tr><td>nTariffIndex</td><td>Int</td><td>1-99</td><td>Current Tariff index</td></tr>
<tr><td>nKeyVersion</td><td>Int</td><td>1-9</td><td>Current Key version</td></tr>
<tr><td>nKeyExpiredTime</td><td>Int</td><td>0-255</td><td>Current Key expire time</td></tr>
<tr><td>nKeyNo</td><td>Int</td><td>0-65535</td><td>Current Key Sequence</td></tr>
</table>
</td></tr>
<tr><td>Return value</td><td>Return XML String
Xml format:
<?xml version="1.0" encoding="UTF-8"?></td></tr>
</table>

	<result> <errorCode></errorCode> <tokens> <token></token> ... <token></token> <tokens> </result> Note：errorCode definition refer to chart 3. If succeeded, need return an array of <tokens>elements

Function definition	public String getKeyChangeToken (String meterNo, String sgc, int tarrifIndex, int keyVersion, int keyExpiredTime, long keyNo, String nSgc, int nTariffIndex, int nKeyVersion,int nKeyExpiredTime, int nKeyNo)
	{
	…
	}

3.2.4 Management Token

3.2.4.1 Generate Clear Meter Balance Token

Functionality	To get clear balance Token
Application	To clear balance
Function name	getClearBalanceToken
Parameter list	

Parameter name	Type	Scope or length	Description
meterNo	String	String of 12 numbers	Meterno.
sgc	String	String of 6 numbers	Supply group code
tariffIndex	Int	1-99	Tariff index
keyVersion	int	1-9	Key version
keyExpiredTime	int	0-255	Key expire time
keyNo	long	0-65535	Key Sequence
seqNo	int	1-255	Token Sequence

Return value	Return XML String Xml format: <?xml version="1.0" encoding="UTF-8"?>

	<result> <errorCode></errorCode> <tokens> <token></token> ... <token></token> <tokens> </result> Note：errorCode definition refer to chart 3. If succeeded, need return an array of <tokens>elements

Function definition	public String getClearBalanceToken (String meterNo, String Sgc, int tarrifIndex,
	int keyVersion, int keyExpiredTime, long keyNo, int seqNo)
	{
	…
	}

3.2.4.2 Generate Clear Event Token

Functionality	To get clear event Token			
Application	To clear event			
Function name	**getClearEventToken**			
Parameter list	Parameter name	Type	Scope or length	Description
	meterNo	String	String of 12 numbers	Meterno.
	sgc	String	String of 6 numbers	Supply group code
	tariffIndex	int	1-99	Tariff index
	keyVersion	int	1-9	Key version
	keyExpiredTime	int	0-255	Key expire time
	keyNo	long	0-65535	Key Sequence
	seqNo	int	1-255	Token Sequence
Return value	Return XML String			

	Xml format: <?xml version="1.0" encoding="UTF-8"?> <result> <errorCode></errorCode> <tokens> <token></token> ... <token></token> <tokens> </result> Note： errorCode definition refer to chart 3. If succeeded, need return an array of <tokens>elements

Function definition	public String getClearEventToken (String meterNo, String Sgc, int tarrifIndex,
	int keyVersion, int keyExpiredTime, long keyNo, int seqNo)
	{
	…
	}

3.2.4.3 Generate Setup MaxPowerLimitModeToken

Functionality	To get the set up MaxPowerLimitModeToken
Application	To set up MaxPowerLimitMode Token
Function name	**getMaxPowerLimitToken**

Parameter list

Parameter name	Type	Scope or length	Description
meterno.	String	String of 12 numbers	Meterno.
sgc	String	String of 6 numbers	Supply group code
tariffIndex	Int	1-99	Tariff index
keyVersion	Int	1-9	Key version
keyExpiredTime	Int	0-255	Key expire time
keyNo	Long	0-65535	Key Sequence

<table>
<tr><td></td><td>
<table>
<tr><td>seq</td><td>Int</td><td>1-255</td><td>Token Sequence</td></tr>
<tr><td>activatingModel</td><td>Int</td><td>0-1
0： Noramal Mode
1： Immediately switch mode</td><td>Active Mode
This parameter always 1.</td></tr>
<tr><td>activatingDate</td><td>String</td><td>YYYY-MM-DD</td><td>Active Date
Change Date type to String type： adapt to all kinds development languages</td></tr>
<tr><td>maxPowerLimits</td><td>int[]</td><td>1-2047 (array length :2) scale&unit:0.1kW</td><td>Max Power array</td></tr>
<tr><td>Hours</td><td>Int[]</td><td>0-23 (array length :2)unit:hour</td><td>Active time</td></tr>
</table>
</td></tr>
<tr><td>Return value</td><td>Return XML String
Xml format:
<?xml version="1.0" encoding="UTF-8"?>
<result>
<errorCode></errorCode>
<tokens>
<token></token>
...
<token></token>
<tokens>
</result>

Note： errorCode definition refer to chart 3.

If succeeded, need return an array of <tokens>elements</td></tr>
<tr><td>Function definition</td><td>public String getMaxPowerLimitToken (String meterNo,
String Sgc,int tarrifIndex, int keyVersion, int keyExpiredTime , long keyNo, int seq , int activatingModel, String date int[] maxPowerLimits, int[] hours)</td></tr>
</table>

	{
	…
	}

3.2.4.4 Generate Setup Single Tariff Token

Functionality	To get the Setup Single Tarrif Token			
Application	To set up single tarrif			
Function name	**getSingleTariffToken**			
Parameter list	Parameter name	Type	Scope or length	Description
	meterNo	String	String of 12 numbers	Meterno.
	sgc	String	String of 6 numbers	Supply group code
	tariffIndex	Int	1-99	Tariff index
	keyVersion	Int	1-9	Key version
	keyExpiredTime	Int	0-255	Key expire time
	keyNo	Long	0-65535	Key Sequence
	seq	Int	1-255	Token Sequence
	activatingDate	String	YYYY-MM-DD	Active date Change Date type to String type：adapt to all kinds development languages
	activatingModel	Int	0-1 0：Normal Mode 1：Immediately switch mode	Active Mode always 1.
	validate	Int	0-7	Validate always 0.
	rate	Int	1-8191（scale&unit：0.01TK）By the integer bits and	Unit price

			fractional bits, up to two decimal places, the maximum rate of price can be set to 81.91	

Return value	Return XML String Xml format: <?xml version="1.0" encoding="UTF-8"?> <result> <errorCode></errorCode> <tokens> <token></token> ... <token></token> <tokens> </result> Note：errorCode definition refer to chart 3. If succeeded, need return an array of <tokens>elements

Function definition	public String getSingleTariffToken(String meterNo, String Sgc, int tarrifIndex,
	int keyVersion, int keyExpiredTime , long keyNo, int seq,
	String activatingDate, int activatingModel,int validDate, int rate)
	{
	…
	}

3.2.4.5 Generate Setup Step Tariff Token

Functionality	To get the Setup Step Tarrif Token			
Application	To set up step tarrif			
Function name	**getStepTariffToken**			
Parameter list	Parameter name	Type	Scope or length	Description
	meterNo	String	String of 12 numbers	Meterno.

	sgc	String	String of 6 numbers	Supply group code
	tariffIndex	int	1-99	Tariff index
	keyVersion	int	1-9	Key version
	keyExpiredTime	int	0-255	Key expire time
	keyNo	long	0-65535	Key Sequence
	seq	int	1-255	Token Sequence
	activatingDate	String	YYYY-MM-DD	Active date Change Date type to String type：adapt to all kinds development languages
	validate	int	0-7	Effective date
	activatingModel	Int	0-1 0：Normal Mode 1：Immediately switch mode	Effective Mode Always 1
	rates	int[]	Length 4 1-8191(scale&unit：0.01TK) By the integer bits and fractional bits, up to two decimal places, the maximum rate of price can be set to 81.91	Price array
	steps	int[]	1-4096 (the max array length :8) scale&unit:1 kWh	Step array
Return value	Return XML String Xml format: <?xml version="1.0" encoding="UTF-8"?> <result> <errorCode></errorCode> <tokens> <token></token>			

	... <token></token> <tokens> </result> **for example：** 0-100 kWh rate：2.3TK 100-200kWh rate: 2.7TK 200-300kWh rate: 3.2TK 300-∞ rate: 4.5TK rate[]= {2.3,2.7,3.2,4.5} steps[]={100,200,300} If succeeded, need return an array of <tokens>elements

Function definition	public String getStepTariffToken(String meterNo, String Sgc, int tarrifIndex,int keyVersion, int keyExpiredTime, long keyNo, int seq,String activatingDate, int activatingModel, int validDate, int[] rates, int[] steps)
	{
	…
	}

3.2.4.6 Generate Setup TOU Tariff Token

Functionality	To get the Setup TOU tariff Token
Application	To set up TOU tariff
Function name	**getTOUTariffToken**
Parameter list	

Parameter name	Type	Scope or length	Description
meterNo.	String	String of 12 numbers	Meterno.
Sgc	String	String of 6 numbers	Supply group code
tariffIndex	Int	1-99	Tariff index
keyVersion	Int	1-9	Key version

	keyExpiredTime	Int	0-255	Key expire time
	keyNo	long	0-65535	Key Sequence
	Seq	int	1-255	Token Sequence
	activatingDate	String	YYYY-MM-DD	Active Date Change Date type to String type：adapt to all kinds development languages
	activatingModel	Int	0-1 0：Normal 1：Immediately switch mode	Active Mode always 1.
	validate	Int	0-7	Effective Date
	rates	Int[]	Length 4 1-8191（scale&unit：0.01TK） By the integer bits and fractional bits, up to two decimal places, the maximum rate of price can be set to 81.91	Price
	times	int[]	Length 2 0-23	Hours period
Return value	Return XML String Xml format: <?xml version="1.0" encoding="UTF-8"?> <result> <errorCode></errorCode> <tokens> <token></token>			

	... <token></token> <tokens> </result> Note：errorCode definition refer to chart 3. If succeeded, need return an array of <tokens>elements

Function definition	public String getTOUTariffToken(String meterNo, String Sgc, int tarrifIndex,int keyVersion, int keyExpiredTime, long keyNo, int seq,String activatingDate, int activatingModel, int validDate, int[] rates, int[] times)
	{
	…
	}

3.2.4.7 Generate Setup friendly mode Token

Functionality	To get the Setup Friendly Mode Token			
Application	To set up friendly mode			
Function name	**getFriendModeToken**			
Parameter list	Parameter name	Type	Scope or length	Description
	meterNo	String	String of 12 numbers	Meterno.
	Sgc	String	String of 6 numbers	Supply group code
	tariffIndex	int	1-99	Tariff index
	keyVersion	int	1-9	Key version
	keyExpiredTime	int	0-255	Key expire time
	keyNo	long	0-65535	Key Sequence
	seqNo	int	1-255	Token Sequence
	friendMode	int	0-1 0: friendly mode is enable 1:friendly mode is closed	Friendly mode

<table>
<tr><td></td><td><table>
<tr><td>Times</td><td>int[]</td><td>0-23
(array length :2)</td><td>Friendly time interval</td></tr>
<tr><td>Days</td><td>int[]</td><td>0-6 means Saturdy, Friday, Thursday , Wednesday, Tuesday , Monday and Sunday in sequence.
(array length :7)</td><td>Weekly holiday
Array value 0 means weekend and 1 means ordinary day</td></tr>
<tr><td>N_of_allowable_days</td><td>int</td><td>0-99</td><td>No of allowable days that friendly hour will work after that without vending meter will not work in friendly hour</td></tr>
</table></td></tr>
<tr><td>Return value</td><td>Return XML String
Xml format:
<?xml version="1.0" encoding="UTF-8"?>
<result>
<errorCode></errorCode>
<tokens>
<token></token>
...
<token></token>
<tokens>
</result>
Note：errorCode definition refer to chart 3.
If succeeded, need return an array of <tokens>elements</td></tr>
<tr><td>Function definition</td><td>public String getFriendModeToken(String meterNo, String Sgc, int tarrifIndex,
int keyVersion, int keyExpiredTime , long keyNo, int seqNo, int friendMode, int[] times, int[] days)
{
…
}</td></tr>
</table>

3.2.4.8 Generate Switch Meter Mode Token

<table>
<tr><td>Functionality</td><td>To get Switch Meter Mode Token</td></tr>
<tr><td>Application</td><td>To set up switch meter mode</td></tr>
<tr><td>Function name</td><td>getChangeMeterModeToken</td></tr>
<tr><td>Parameter list</td><td>
<table>
<tr><th>Parameter name</th><th>Type</th><th>Scope or length</th><th>Description</th></tr>
<tr><td>meterNo</td><td>String</td><td>String of 12 numbers</td><td>Meterno.</td></tr>
<tr><td>Sgc</td><td>String</td><td>String of 6 numbers</td><td>Supply group code</td></tr>
<tr><td>tariffIndex</td><td>int</td><td>1-99</td><td>Tariff index</td></tr>
<tr><td>keyVersion</td><td>int</td><td>1-9</td><td>Key version</td></tr>
<tr><td>keyExpiredTime</td><td>int</td><td>0-255</td><td>Key expire time</td></tr>
<tr><td>keyNo</td><td>long</td><td>0-65535</td><td>Key Sequence</td></tr>
<tr><td>seqNo</td><td>int</td><td>1-255</td><td>Token Sequence</td></tr>
<tr><td>mode</td><td>int</td><td>0-1
0:switch to the ordinary meter mode;
1:switch to the pre-paid meter mode</td><td>Meter mode switch</td></tr>
</table>
</td></tr>
<tr><td>Return value</td><td>Return XML String
Xml format:
<?xml version="1.0" encoding="UTF-8"?>
<result>
<errorCode></errorCode>
<tokens>
<token></token>
...
<token></token>
<tokens>
</result>
Note： errorCode definition refer to chart 3.
If succeeded, need return an array of <tokens>elements</td></tr>
<tr><td>Function</td><td>public String getChangeMeterModeToken(String meterNo, String Sgc,</td></tr>
</table>

definition	int tarrifIndex, int keyVersion, int keyExpiredTime, long keyNo, int seqNo, int mode)
	{
	…
	}

3.2.4.9 Generate SetCreditAmountLimitAndOverdrawAmountLimitToken

<table>
<tr><td>Functionality</td><td>To get set credit amount limit and overdrawAmountLimit Token</td></tr>
<tr><td>Application</td><td>To set credit amount limit and overdrawAmountLimit.</td></tr>
<tr><td>Function name</td><td>getSetCreditAmountLimitAndOverdrawAmountLimit</td></tr>
<tr><td>Parameter list</td><td>
<table>
<tr><th>Parameter name</th><th>Type</th><th>Scope or length</th><th>Description</th></tr>
<tr><td>meterNo</td><td>String</td><td>String of 12 numbers</td><td>Meterno.</td></tr>
<tr><td>Sgc</td><td>String</td><td>String of 6 numbers</td><td>Supply group code</td></tr>
<tr><td>tariffIndex</td><td>Int</td><td>1-99</td><td>Tariff index</td></tr>
<tr><td>keyVersion</td><td>Int</td><td>1-9</td><td>Key version</td></tr>
<tr><td>keyExpiredTime</td><td>Int</td><td>0-255</td><td>Key expire time</td></tr>
<tr><td>keyNo</td><td>Long</td><td>0-65535</td><td>Key Sequence</td></tr>
<tr><td>seqNo</td><td>Int</td><td>1-255</td><td>Token Sequence</td></tr>
<tr><td>amountType</td><td>Int</td><td>0-1
0: creditAmountLimit
1: overAmountLimit</td><td>Balance Type</td></tr>
<tr><td>amountLimit</td><td>Int</td><td>0~99999999
(scale&unit:0.01TK)</td><td>Amount</td></tr>
</table>
</td></tr>
<tr><td>Return value</td><td>Return XML String
Xml format:
<?xml version="1.0" encoding="UTF-8"?>
<result>
<errorCode></errorCode>
<tokens>
<token></token></td></tr>
</table>

	... <token></token> <tokens> </result> Note：errorCode definition refer to chart 3. If succeeded, need return an array of <tokens>elements

Function definition	public String getSetCreditAmountLimitOrOverdrawAmountLimitToken (
	String meterNo, String Sgc, int tarrifIndex, int keyVersion, int keyExpiredTime, long keyNo, int seqNo, int amountType, int amountLimit)
	{
	…
	}

3.2.4.10 Generate Logoff Token

Functionality	To get reset Token			
Application	To reset meter			
Function name	**generateLogoffReturnToken**			
Parameter list	Parameter name	Type	Scope or length	Description
	meterNo	String	String of 12 numbers	Meterno.
	Sgc	String	String of 6 numbers	Supply group code
	tariffIndex	Int	1-99	Tariff index
	keyVersion	Int	1-9	Key version
	keyExpiredTime	Int	0-255	Key expire time
	keyNo	Long	0-65535	Key Sequence
	seqNo	Int	1-255	Token Sequence

Return value	Return XML String Xml format: <?xml version="1.0" encoding="UTF-8"?> <result> <errorCode></errorCode> <tokens> <token></token> ... <token></token> <tokens> </result> Note：errorCode definition refer to chart 3. If succeeded, need return an array of <tokens>elements

Function definition	public String generateLogoffReturnToken (String meterNo, String Sgc, int tarrifIndex, int keyVersion, int keyExpiredTime, int keyNo, int seqNo)
	{
	…
	}

3.2.4.11 Generate Credit/ Management Return Token

Functionality	To get return Credit/Management Return Token
Application	User should get the return token before credit Return ,and send the meter information to the electricity sale system, or else the electricity sale system will not sale electricity to user ,it will helpful for subsequent management.
Function name	**resolveReturnToke**
Parameter list	

Parameter name	Type	Scope or length	Description
meterNo	String	String of 12 numbers	Meterno.
Sgc	String	String of 6 numbers	SGCID
tariffIndex	int	1-99	Tariff index

	keyVersion	int	1-9	Key version
	keyExpiredTime	int	0-255	Key expire time
	keyNo	long	0-65535	Key Sequence
	Token	String	Custom defined by each manufacturers	Token encryption character String

Return value

Return XML String

Xml format:

```
<?xml version='1.0' encoding='utf-8'?>
<result>
    <type>0</type>
    <balance>454.54</balance>
    <sequence>6</sequence>
    <event>
        <clockSetFlag>1</clockSetFlag>
        <batteryVoltageLowFlag>0</batteryVoltageLowFlag>
        <openCoverFlag>0</openCoverFlag>
        <openBottomCoverFlag>1</openBottomCoverFlag>
        <byPassFlag>0</byPassFlag>
        <reverseFlag>0</reverseFlag>
        <magneticInterfereFlag>0</magneticInterfereFlag>
        <relayStatusFlag>1</relayStatusFlag>
        <relayFaultFlag>0</relayFaultFlag>
        <overdraftUsedFlag>0</overdraftUsedFlag>
        <forwardActiveEnergyTol>11.23</forwardActiveEnergyTol>
        <tariffIndex>51</tariffIndex>
    </event>
</result>
```

Note:If token resolve fail then return null.

Return returnTokens:

Parameter	Scope of Length	Description
Type	0-1	

	0:credit return token 1:logoff return token	
Balance		Balance of meter
Sequence	0-1 0:credit return token 1:logoff return token	Sequence of meter
clockSetFlag	0-1 0:unhappen 1:happen	Clock set event
batteryVoltageLowFlag	0-1 0:unhappen 1:happen	Battery low voltage event
openCoverFlag	0-1 0:unhappen 1:happen	Meter cover open event
openBottomCoverFlag	0-1 0:unhappen 1:happen	Terminal cover open event
byPassFlag	0-1 0:unhappen 1:happen	Bypass event
reverseFlag	0-1 0:unhappen 1:happen	Reverse event
magneticInterfereFlag	0-1 0:unhappen 1:happen	Magnetic interference event
relayStatusFlag	0-1 0:connect 1:disconnect	Relay status

	relayFaultFlag	0-1 0:unhappen 1:happen	Relay fault event
	overdraftUsedFlag	0-1 0:no 1:yes	Emergency balance used or not

Function definition	public String resolveReturnToken (String meterNo, String Sgc, int tarrifIndex, int keyVersion, int keyExpiredTime, long keyNo, String Token)
	{
	…
	}		

3.2.5 Test Function Interface Define

Functionality	To get test Token
Application	To manage the user meter test and display, call the function and generate the test token.
Function name	**getTestToken**
Parameter list	

Parameter name	Type	Scope or length	Description
ManufacturingID	Int	2 numbers	Manufacturer code Hexing:14
control	Int	0 : test all the contents 1 : test relay 2 : test LCD display 3 : teat total energy 4 : test max power limit 5 : display current meter status 6 : display current power 7： display meter version number 8： display current tariff unit price 9： display overload current limit 10: display credit numbers 11: display serial number of meter 12: display off numbers of relay 13: enter into accuracy test mode 18-36: save	Test token type

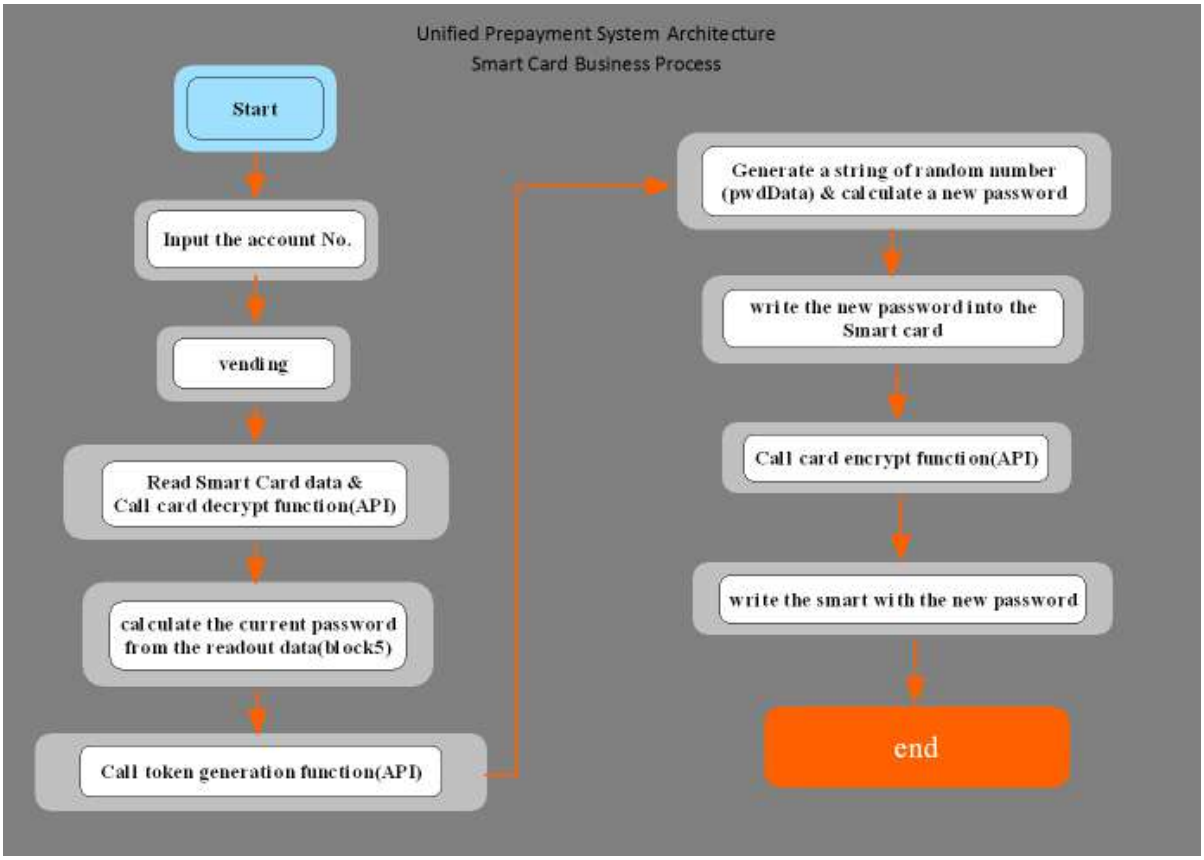
Return value	Return XML String Xml format: <?xml version="1.0" encoding="UTF-8"?>

	<result> <errorCode></errorCode> <tokens> <token></token> ... <token></token> <tokens> </result> Note： errorCode definition refer to chart 3. If succeeded, need return an array of <tokens>elements

Function definition	public String getTestToken (int manufacturingID, int control)
	{
	…
	}

4 Functionality of Read and Write With Card

Smart Cardbusiness process



4.1 Smart Card Data Format

Data	***Bytes***	***Memory Address (Bytes)***
Non Encrypted Data		

Answer To Reset	4	0-4
Binary Pattern	2	4-6
Version	1	6-6
Meter ID	10	7-16
Consumer ID	10	17-26
Utility ID	6	27-32
Sanctioned Load	6	33-38
Meter Type	1	39-39
Sanctioned Load exceeded	2	40-41
Last Recharge Amount	4	42-45
Last Recharge Date	3	46-48
Last Transaction ID	10	49-58
Encrypted Data as Generated by the Meter Manufacturer SDK/API		
The following Data must exist within the encrypted Data: ***Data To Meter:*** Meter ID Consumer ID Utility ID Tokens Credit Token Management Token Test Token ***Data From Meter:*** Tamper Status RTC Status LR Status Usage Data for the last 6 months individually Month Year KWh Taka Recharged Taka Used Average Power Reactive Power Maximum Power Volt Ampere KWh in Peak KWh in Offpeak KVarh in Peak KVarh in Offpeak Total charge in Peak Total charge in Offpeak Number of power failures Number of time sanctioned load exceeded Tamper Number of times tampered Date/Time of Tamper	940	100-1024

Retrun Token		

5 GPRS Meters Connection Platform Design

5.1Concept Description of MIC Communication Access

Smart GPRS meters communicate with MIC using XML format based on TCP. It will use encryption and digital signature technical in TCP linker layer, and in data area of XML document, follow the standard DLMS protocol.

XML Communication Protocol Define

XML Format Define

<?xml version='1.0'?>
<h:rt xmlns h='ProtocolHead'>
<h:pv>1</h:pv>
<h:addr>030140000170</h:addr>
<h:dir>down</h:dir>
<h:pt>1</h:pt>
<h:fc>3</h:fc>
<h:seq>7</h:seq>
<h:e>10</h:e>
<h:a>0</h:a>
<h:r>485845110000000000000007</h:r>< h:d>
0A15CEDF16.....0A15CEDF16
</h:d>
<h:sg>0203<h:sg>
</h:rt>

XML Label Description：

All data of XML should be transferred using ASCII code.

Protocol Presentation	Protocol Description	Note
<? xml version='1.0' ?>	XMLfixed head format	
<h:rt xmlns h='ProtocolHead'>	To name the space under root element.Protocol Head means the first layer of protocol.	
<h:pv>1</ h:pv>	XMLprotocol internal version number of company. 1: the first version	
< h:f>100000000</h:f>	Address of transmit end， the label means which transmit end the data come from. The address will be 1 if the transmit end is	

	master station; The address will be meter address if the transmit end is meter, whose length is not fixed and the max length is 32 pcs ASCII code.	
<h:dir>down</h:dir>	The direction of transmission. up: meter to server down: server to meter	
<h:pt>1</h:pt >	Data transport protocol in data area. 1：DLMS transport protocol 2: XML transport protocol（Company will develop in the future, as the second stage）	
<h:fc>1</h:fc>	Function code: it is mainly for front-end processor distinguishes the data types. When front-end processor doesn't analysis the data or decrypt the encrypted data, it can response the frame transmitted from the meters. At the same time, the label can distinguish who is the transmit end. 1：Link interface test(meter launches) 2：Link interface test response frame(master station responses) 3：Request – response frame（master station responses） 4：Request —confirm/deny frame（master station responses） 5：Report initially- confirm/deny frame（meter launches） 6：Report initially confirm frame needed（meter launches） 7：Report initially confirm frame needless（meter launches） 8：business password refresh（master station launches） 9：manufacturer request（mainly for upgrade remotely）（master station launches） 10：manufacturer respond （mainly for upgrade remotely））（master station launches）	
< h:seq>3< h:seq>	XML frame serial number，for defining which response frame was responded by the response end. The frame serial number is changed by primary end, it will add 1 when primary end launches one frame of message. The range of variationcan be0-99. Driven end regards the frame serial number of primary end as the response frame serial number. After messages transmitted by starting end, when the responded can't be received timely , the serial number of retransmission will be same if the starting end allows retransmission, and the max retransmission numbers is 3; 2） If the driven end receives two starting frame with the same serial number in series, it means the response hasn't been received yet, then retransmission(and without dealing with the message again); 3） If starting end receives two response frames with the same serial number in series, then will not deal with the second response frame.	

< h:e>10</h:e>	Data encryption algorithm 00： without using encryption algorithm ， data transmitted using plaintext form 10：AES-128 In GCM Mode 11：AES-192 12：AES-256 20：DES 30：SHA 40：ECC	
< h:a>1</h:a>	Verify algorithm 0： without using verify 1：SHA-256	
< h:r>4827762</h:r>	Random number. The starting end will regard the random number as vector to encrypt the data area, the receiving end will regard the random number as vector to decrypt the data area. It has 12 bytes, as 24 bytes HEX character String in XML.	
< h:d >123456< /h:d >	Data area, if without using encryption algorithm, the flag is plaintext ; if using encryption algorithm, the flag is the plaintext will be encrypted.	
<h:sg>0203<h:sg>	When without using verify, the flag is useless; when using verify, the flag is verify code for plaintext of data area.	
</h:rt>	XML end flag.	

5.2 Function Code Description：

Link interface test: the function is distinguished in data area.

01：log in

02：quit

03：heartbeat

Link interface test response frame: the function is distinguished in data area.

01：log in

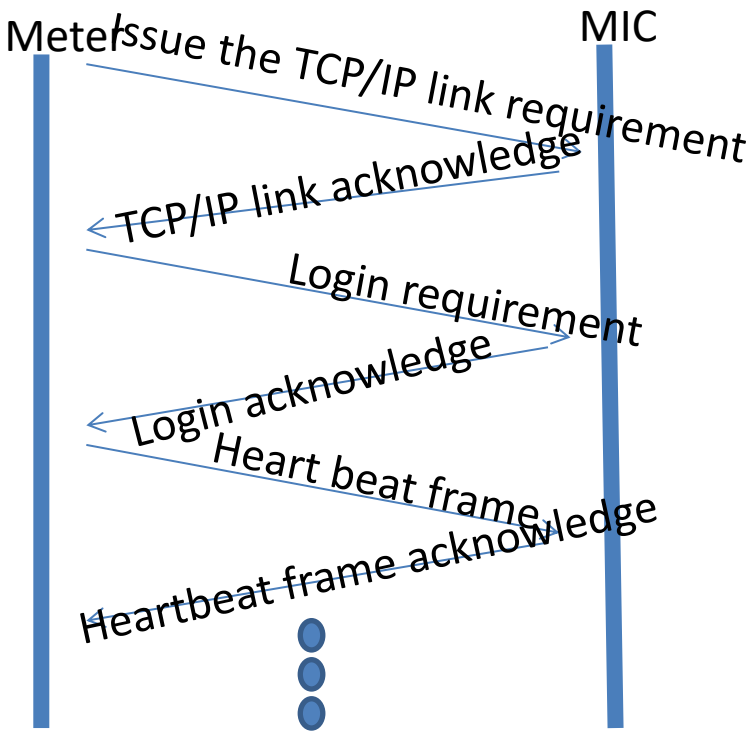
02：quit

03：heartbeat

In this system, meters as client side, MIC as server side, link build , log in request and heatbeat are transmitted by meters , and responded by MIC.

Log in frame: For transmitting a group of specific data to the MIC and confirming the link chaining relationship with master station , after meters connected with master station in TCP/IP.

Heartbeat frame: For transmitting a group of data to maintain the TCP/IP link chaining .



Request – response frame

Request – response frame is the request frame transmitted from master station and the response frame responded by meters.

The correct response means in XML layers, encryption and decryption runs correct or other data to be tested are correct, without including part or all deny for DLMS frames in DLMS layers.

Request- confirm\deny frame

Request- confirm\deny frame is the deny frame responded to the master station when errors happened on data that meters transmitted to the master station.
It will response confirm frame if the update key request to master station is correct.

Request- confirm\deny frame format：

Confirm frame	Function code:00	
Deny frame	Function code:01	Error code： 01: password error; 02: verify error;03: other error

Report initially-confirm\deny frame
For data meters transmitted to the master station , if need confirmation, front-end processor will using the function code to achieveconfirmation and deny for meters.
The format refers to function code 3.
Report initially -confirm frame needed
For data meters reported to the master station initially, if need confirmed , sent according to the function code.
Report initially -confirm frame needless
For data meters reported to the master station initially, if confirmed needless , sent according to the function code.
Business password updated
To update the business password according to the function , sent by master station , encrypted by root key.
Manufacturer request
For self defined commands send from master station to the manufacturers.
Manufacturer response
For self defined commands responded from meters to the manufacturers.
It adopts plaintext transmission mode for data areas of function code 1,2,4,5.

Further information please check<The process of online meter communication >

Annexure –A

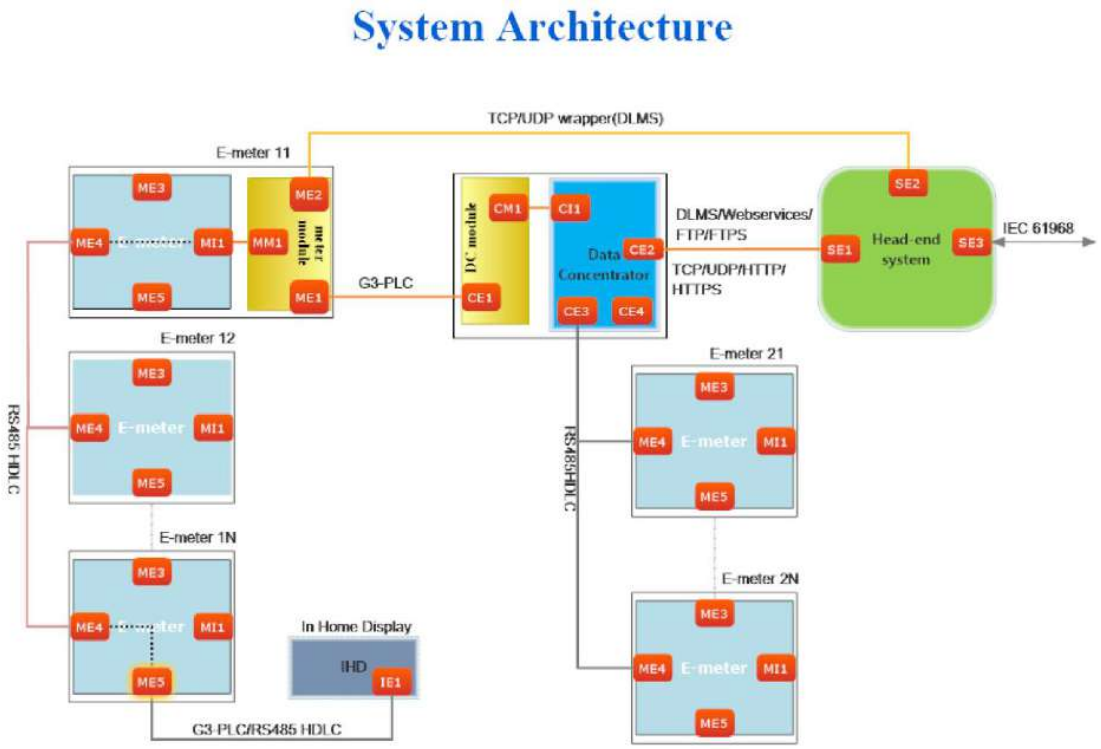
Unified Meter Data Collection System

Addressing System

1. Introduction

1.1 Brief Introduction

The system interface architecture of the device is shown in the following figure; the electric energy meter, concentrator and front end system are simplified into several interfaces marked with correspondent code name respectively;



Device functions & Interoperability Interfacing Requirements are composed of 3 parts, including:

Package 0： Addressing System

Package 1： DCU to HES

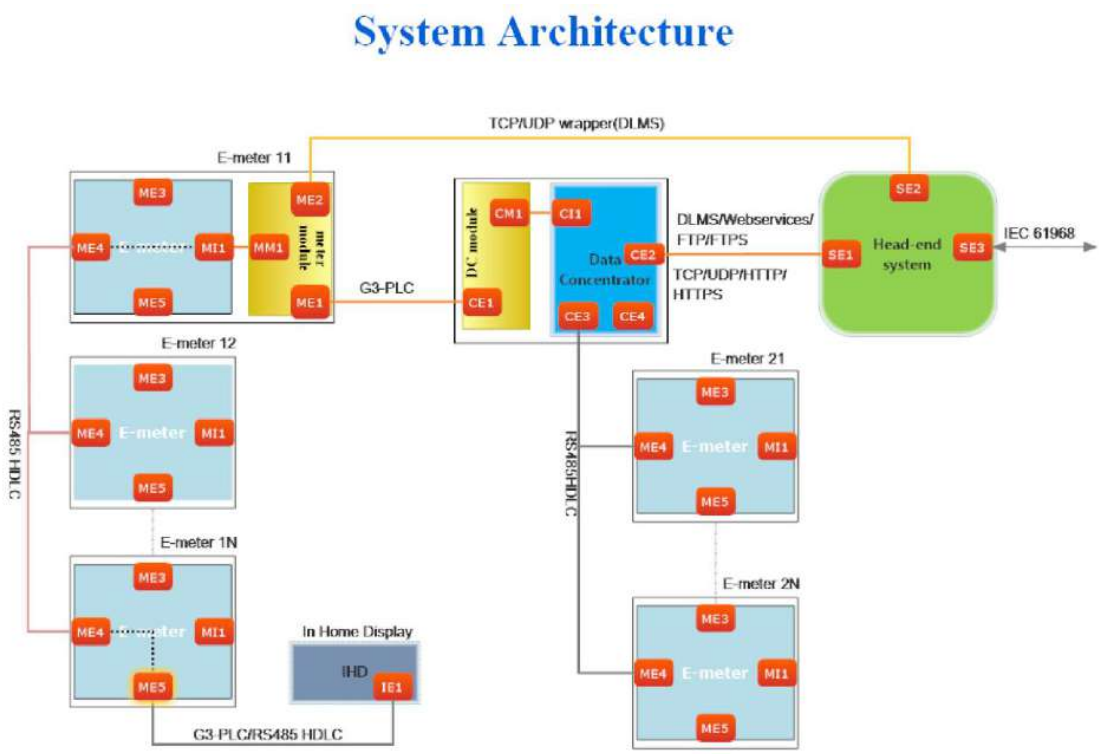
Package 2： Meter Modeling

1.2 Reference Documents

Name
Blue_Book_12th_edition
Green_Book_8th_edition
Object_defs_v3.1_161215
White GlossaryOf DLMS COSEM Terms

2. Communications Architecture

The following diagram describes the communication architecture between HES, concentrator / collector, electricity energy meter and In Home Display unit;



2.1 Description of customer / server side architecture

The data exchange between the data acquisition system and the collected equipment, should refer to the communication mode of the client / server AP specified by DLMS. The client AP initiates the access request and the server AP responds, which is a question and answer communication mechanism; and the role of server AP is specified to be assumed by the meter or module.

By means of access management, one server AP can exchange data with one or more clientAPs at the same time, and a client AP can also access one or more server APs at the same time.

In addition to the normal question and answer mechanism, if there are abnormal events in the server AP, the event can be reported through the prescribed format.

The three-layer architecture of DLMS is shown in the following figure:

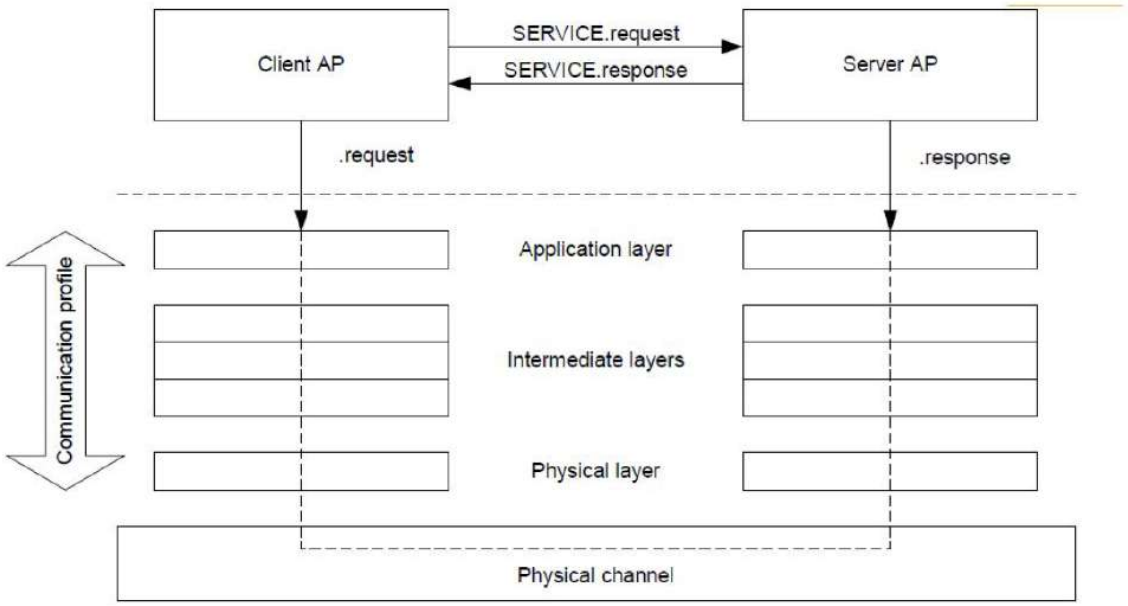


Figure 2 – Client/server relationship and protocols

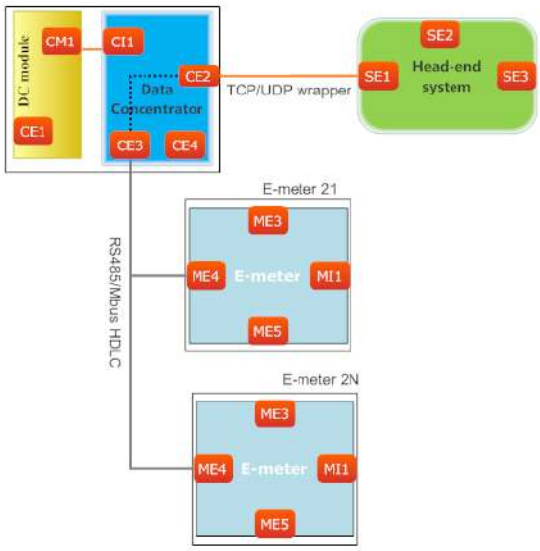
Level 1: Physical Device Layer
Level 2: Logical Device Layer/ Intermediate Layer
Level 3: COSEM Application Layer

2.2 Communication Structure Type

AMI system **client AP-> server AP**, there are six types in communication structure;

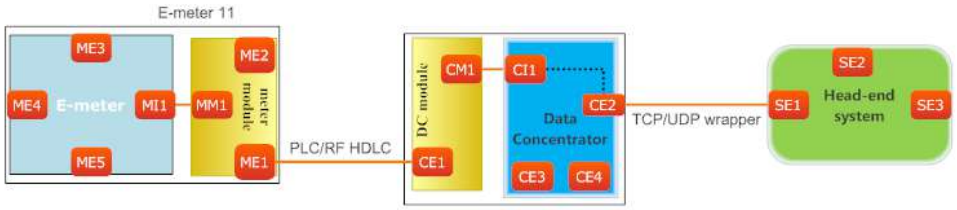
1. SE1(HES)->CE2(Data Concentrator/Collector)->CE3/CE4(Data Concentrator/Collector)->ME4(Electricity Energy Meter COSEM)

Addressing From I



2. SE1(HES)->CE2(Data Concentrator/Collector)->CI1(Data Concentrator/Collector)->CE1(Data Concentrator/Collector) ->ME1(Electricity Energy Meter Communication Module)->MI1(Electricity Energy Meter COSEM)

Addressing From II

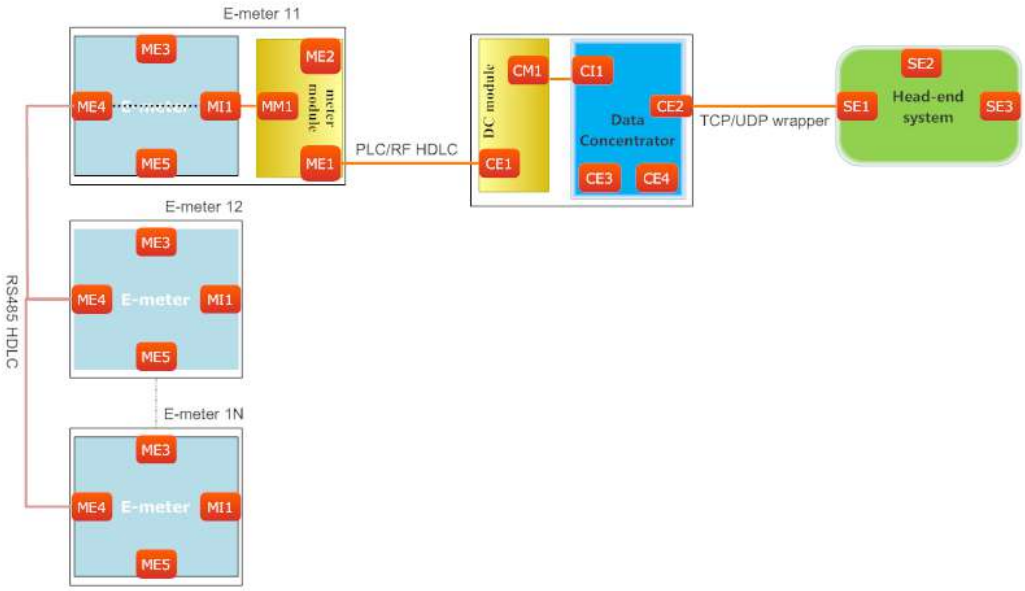


3. SE2(HES)->ME2(Electricity Energy Meter Communication Module)->MI1(Electricity Energy Meter COSEM)

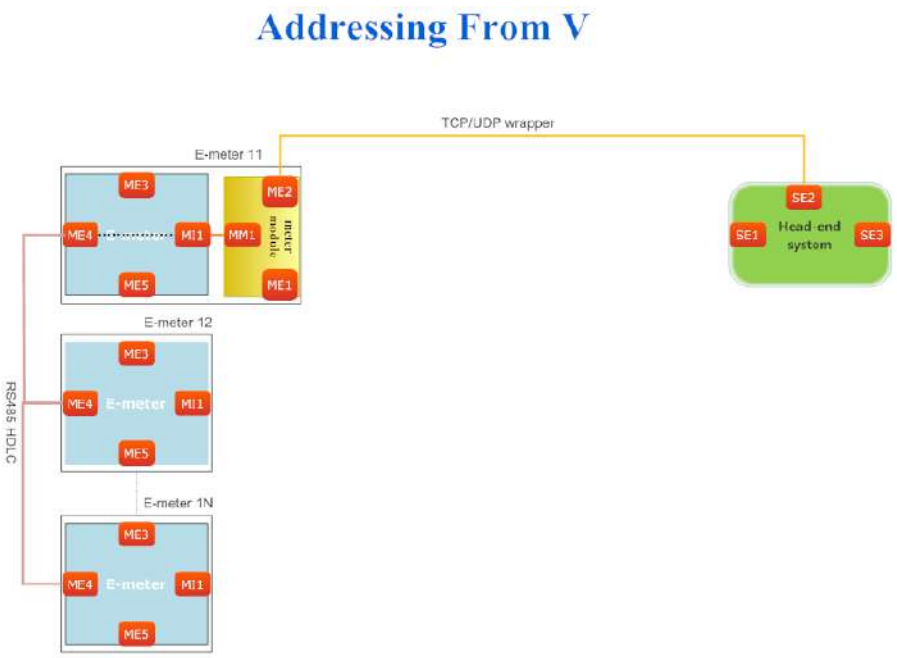
Addressing From III

4. SE1(HES)->CE2(Data Concentrator/Collector)->CI1(Data Concentrator/Collector)->CE1(Data Concentrator/Collector) ->ME1(Electricity Energy Meter Communication Module)->ME4(Electricity Energy Meter COSEM)

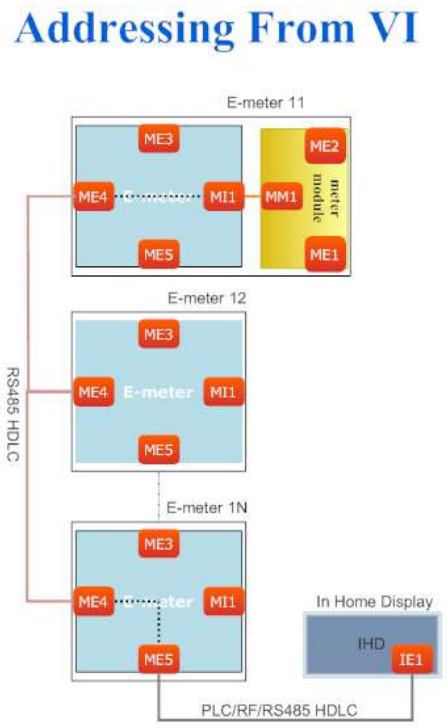
Addressing From IV



5. SE2(HES)->ME2(Electricity Energy Meter Communication Module)->ME4(Electricity Energy Meter COSEM)



6. IE1(IHD)->ME5(Electricity Energy Meter IHD Interface)->ME4(Electricity Energy Meter COSEM)



2.3 Scope and Addressing Rules

As shown in the addressing system diagram, the range of addressing rules defined in this document mainly includes address assignment at the communication interface level of Wrapper, Gateway protocol and HDLC, and the corresponding conversion rules;

As shown in Chapter 2.2, the AMI system supports six types of typical AMI structures; It should be noted that the actual application is not limited to the above six types of form; Depending on the function changes of the device, the specific structure will change. For example, when the IHD can also serve as a server AP, the data can be forwarded through the ME5 interface via the meter by the data push service, and push it to HES;

In order to ensure the interoperability of the equipment and the unity of the system, for the same interface, no matter what kind of AMI structure is adopted, the way of accessing and addressing is the same.

3. Wrapper

3.1 Wrapper Address Format Description

Wrapper includes version, SSAP, DSAP and LEN, a total of 8 bytes in frame format as follows:

■Parameter Description:
Version: 2 bytes, Wrapper version, current version is 0x0001;
SSAP: 2 bytes, represents SAP address of sender;
DSAP: 2 bytes, values range is 0x0001 – 0xFFFF, represents destination address;
LEN: 2 bytes, values range is 0x0000 – 0xFFFF, indicates the byte length of the transmitted COSEM APDU;
COSEM APDU: communication data content, does not belong to the wrapper category.

4. HDLC

4.1 HDLC Address Format Description

The frame format of HDLC is described here only for the address field as follows:

0x7e	Frame Type and Frame Length	Destination Address Field	Source Address Field	Control Domain	Frame Header Verification	LLC Frame Header	User Data Infor-mation	Data Frame Check	0x7e

The address field is divided into two parts, the destination address field and the source address field. For the client AP, the destination address is the address of the accessed device (server AP) and the source address is the client AP address.
For the accessed device, it is opposite when replying to the access frame of the client AP. The destination address is the client AP address, and the source address is the address of the accessed device;
The client AP address is defined as 1 byte; the range of values is: 0x00-0x7F; the definition and reference is in the low bytes of the 4.2.1 SSAP client AP address;
The server AP address can support 1 / 2 / 4 bytes lengths, as shown in the following table:

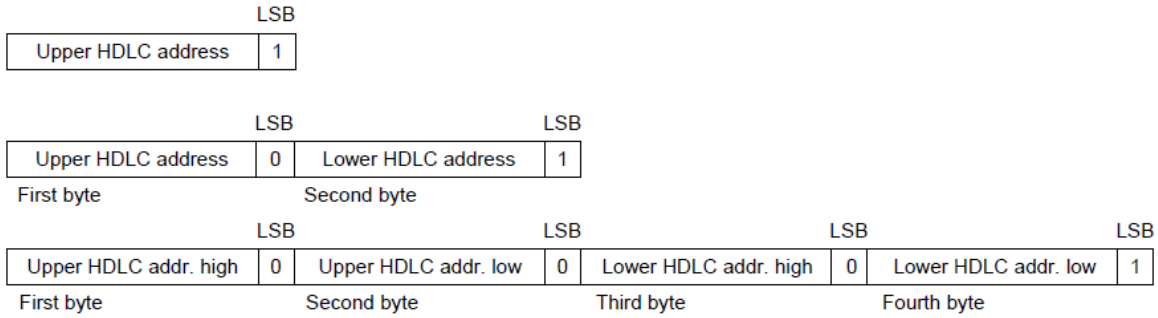


Figure 51 – Valid server address structures

Upper HDLC Address, used to express logical addresses,
Lower HDLC address, used to describe a physical address, 1 or 2 bytes.
Upper HDLC address, used to distinguish different logical devices within the same physical device.
LowerHDLC address cannot appear if not needed. For example, through infrared local access devices, it only needs the default upper HDLC address(0x01) without knowing the physical address.

5. Gateway Protocol

The Gateway protocol is mainly used for gateway devices (concentrator / collector / communication module), transparent forwarding of multilevel networks, as shown in the following figure:

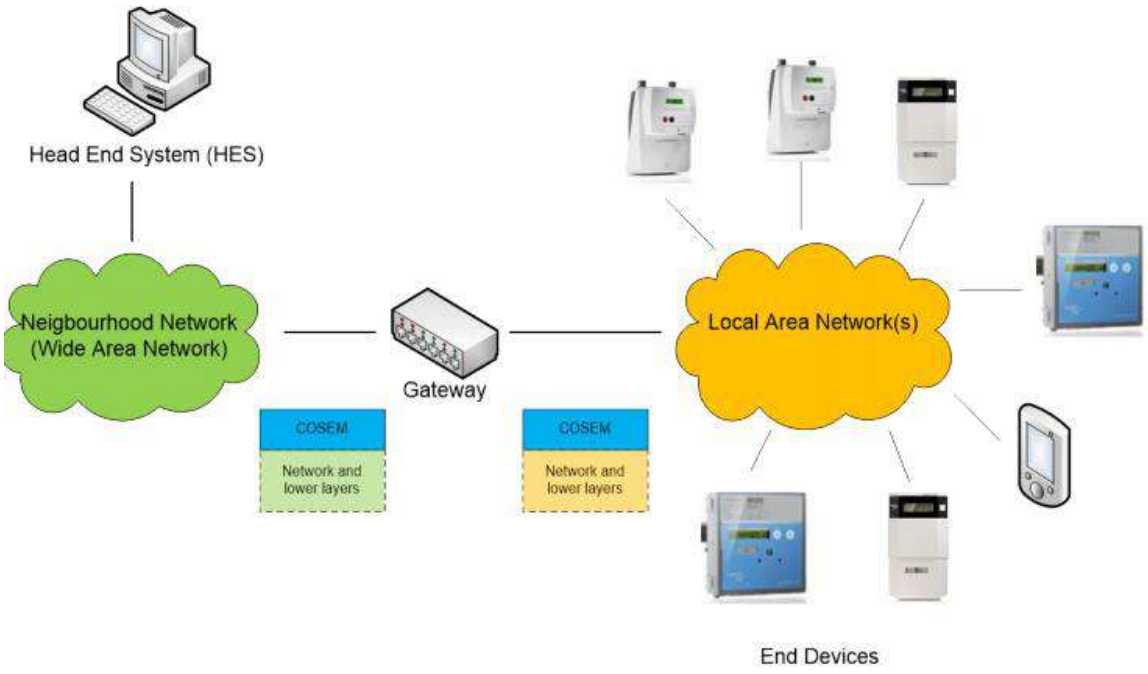


Figure 156 – General architecture with gateway

5.1 Gateway protocol Format Description

The definition of gateway protocol is as follows: Prefix for short;

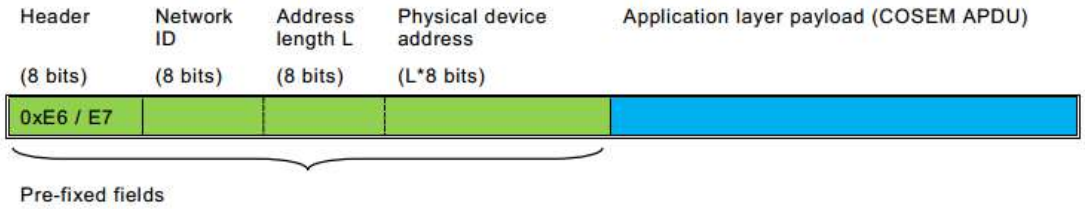


Figure 157 – The fields used for pre-fixing the COSEM APDUs

■Parameter Description:

Header:1 byte, indicating the data transfer direction, 0xE6 is a request or data report frame , 0xE7 is a response frame;

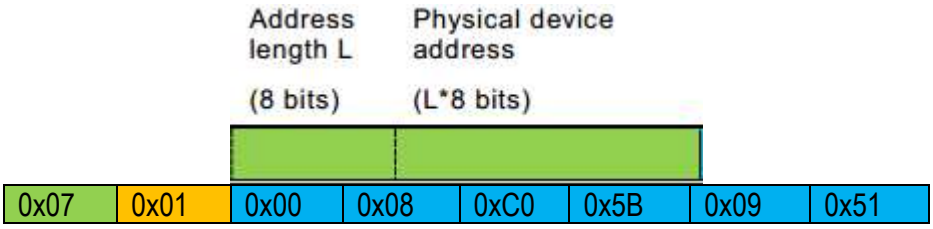
Network ID: 1 byte, network ID, indicating the forwarding of the lower channel number (network number);

Address length: 1 byte, target address length;

Physical device address:N bytes, the physical address of the target,whose length depends on the address length;

COSEM APDU： The communication data content, does not belong to the gateway protocol.

For example: the serial number of the equipment product is: 037586930001, then the physical address of gateway protocol is 0x0008C05B0951;



6. Address Delivery and Conversion Rules

This section mainly defines nesting rules in various level architechture of wrapper, Gateway protocol, HDLC and (APDU)COSEMwhen device is in forwarding and non-forwarding status.

6.1 Data Frame Nested Format

6.1.1 Non-Forwarding Rule

When the destination device is in the same accessed physical device (non-forwarding device), using the following structure to communicate according to the addressing structure in Chapter 2:

1. Wrapper+COSEM（APDU）

Version	SSAP	DSAP	LEN	COSEM APDU

Among which, DSAP： <=0x001F

2. HDLC+COSEM(APDU)

0x7e	Frame Type	Destination	Source	Control	Frame	LLC	COSE	Data	0x7e

	and Frame Length	Address Field	Address Field	Domain	Head Check	Frame Head	M (APDU)	Frame Check	

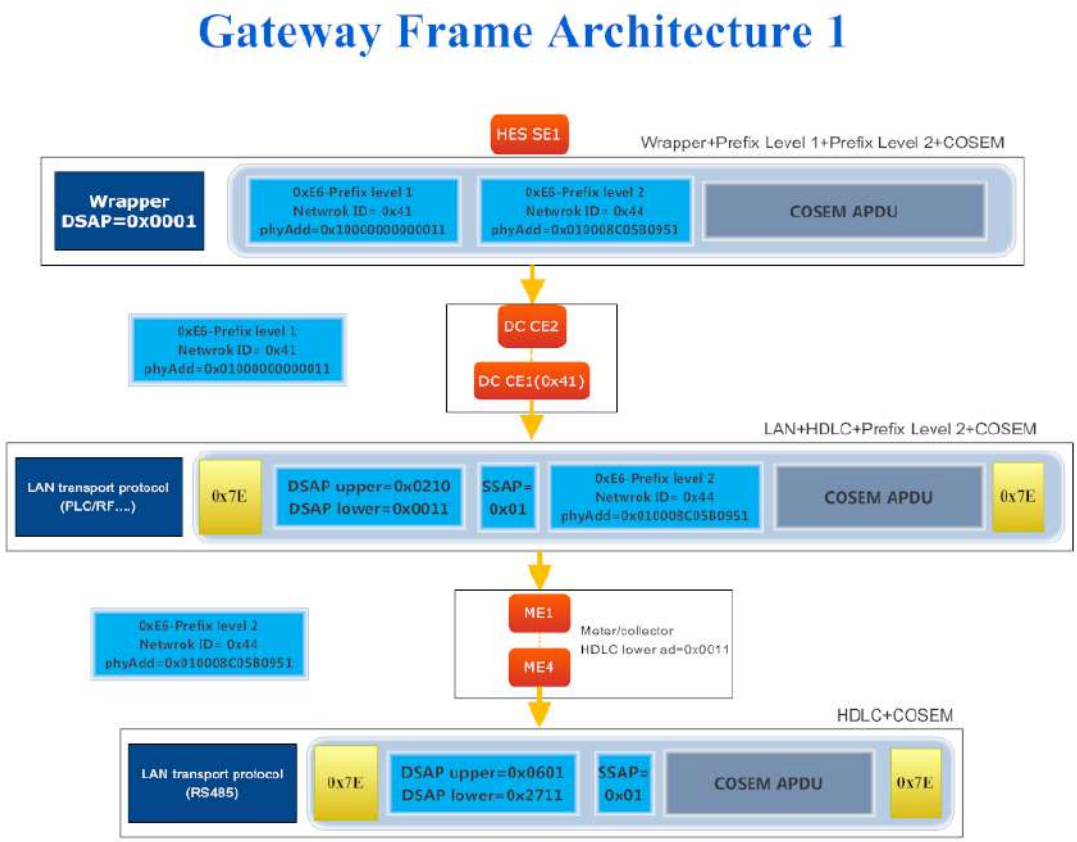
The destination address: upper HDLC address <= 0x001F;

6.1.2Gateway Protocol Forwarding

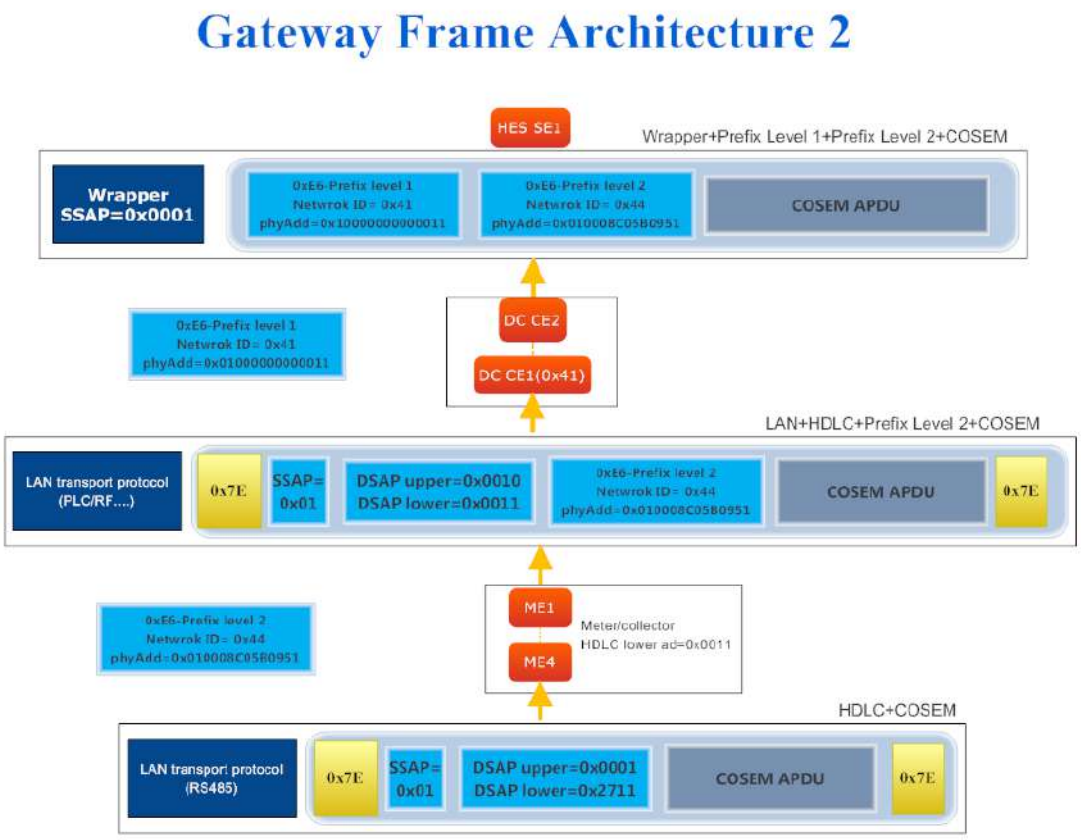
The Gateway forwarding is carried out according to the destination channel （network ID） in prefix and the physical address. In this forwarding protocol, the wrapper and HDLC are not nested, and the data domain content of all transmissions is prefix+APDU(COSEM); According to its own channel, the communication device packages MAC and data link layer for data communication;

6.1.2.1Channel Occupancy Forwarding

Corresponding to the channel occupancy forwarding, the superior equipment task is in the waiting status, and records and forwards the port information in task status and packs and delivers the data to the first level device after the data replies. Then the channel is released



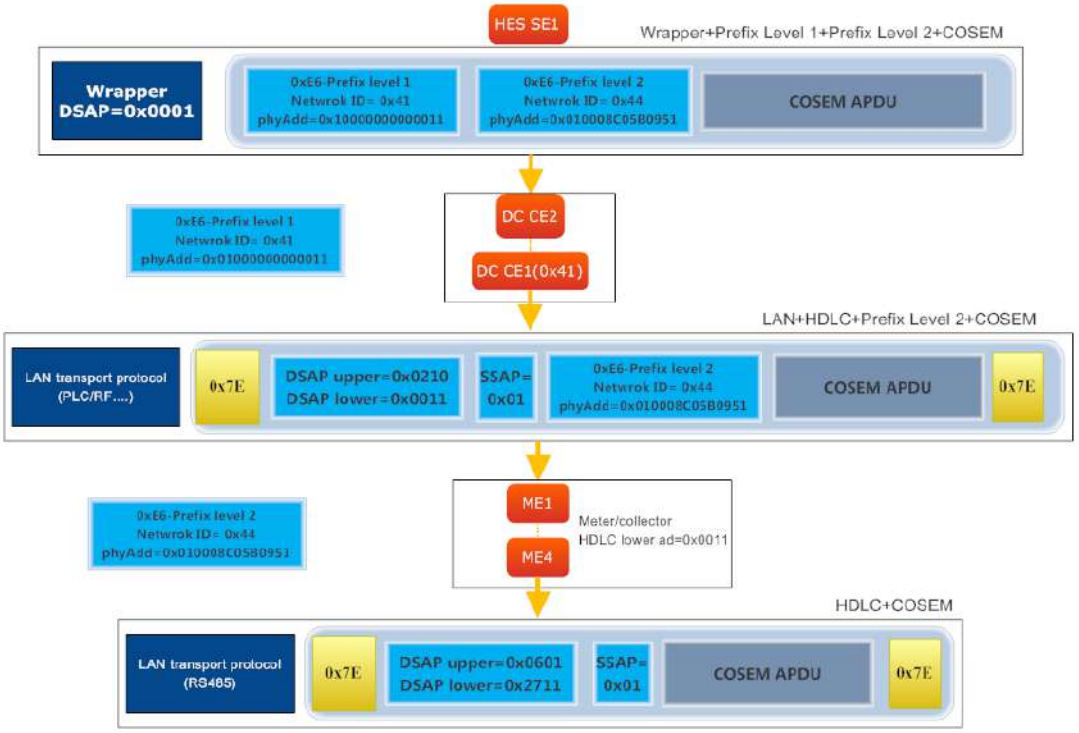
Corresponding to the channel occupancy forwarding, the superior equipment task is in the waiting status, and records and forwards the port information in task status and packs and delivers the data to the first level device after the data replies. Then the channel is released.



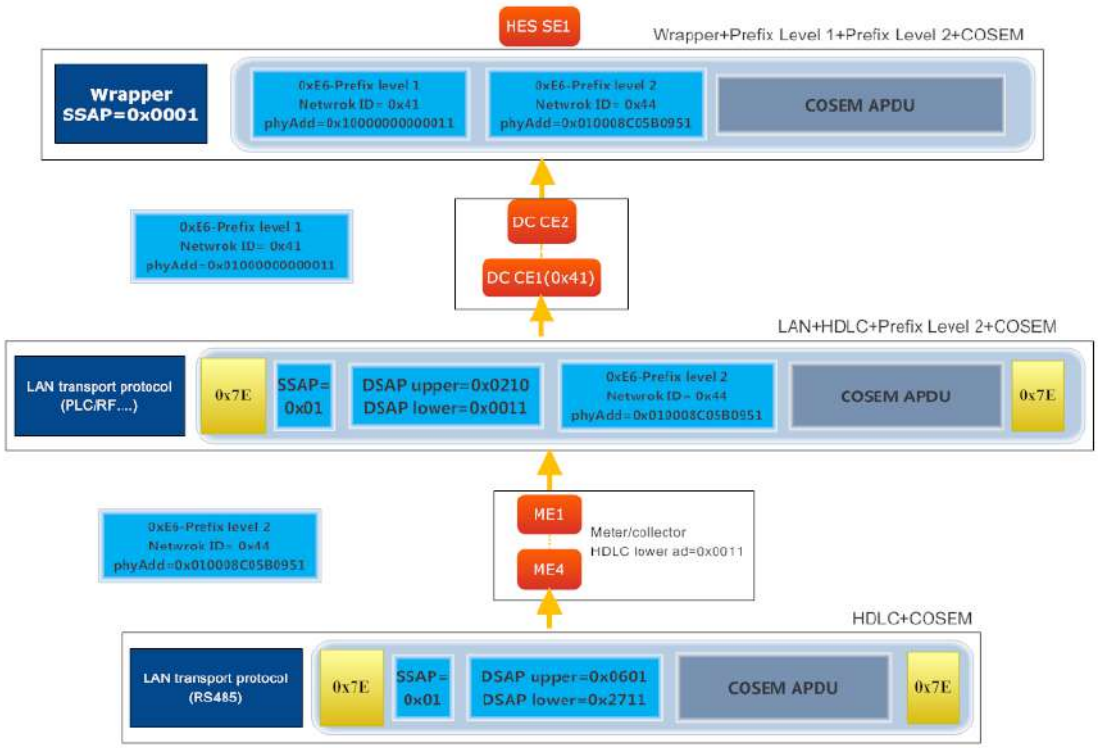
6.1.2.1Address Conversion Forwarding

During the forwarding process, the forwarding port is marked to the next level of HDLC data frame, the forwarding task does not need to pause the current task of the channel passing through, and there is no need to wait for timeout. After the target replies the data, step by step reply will start according to the marked port number of each level;

Gateway Frame Architecture 3



Gateway Frame Architecture 4



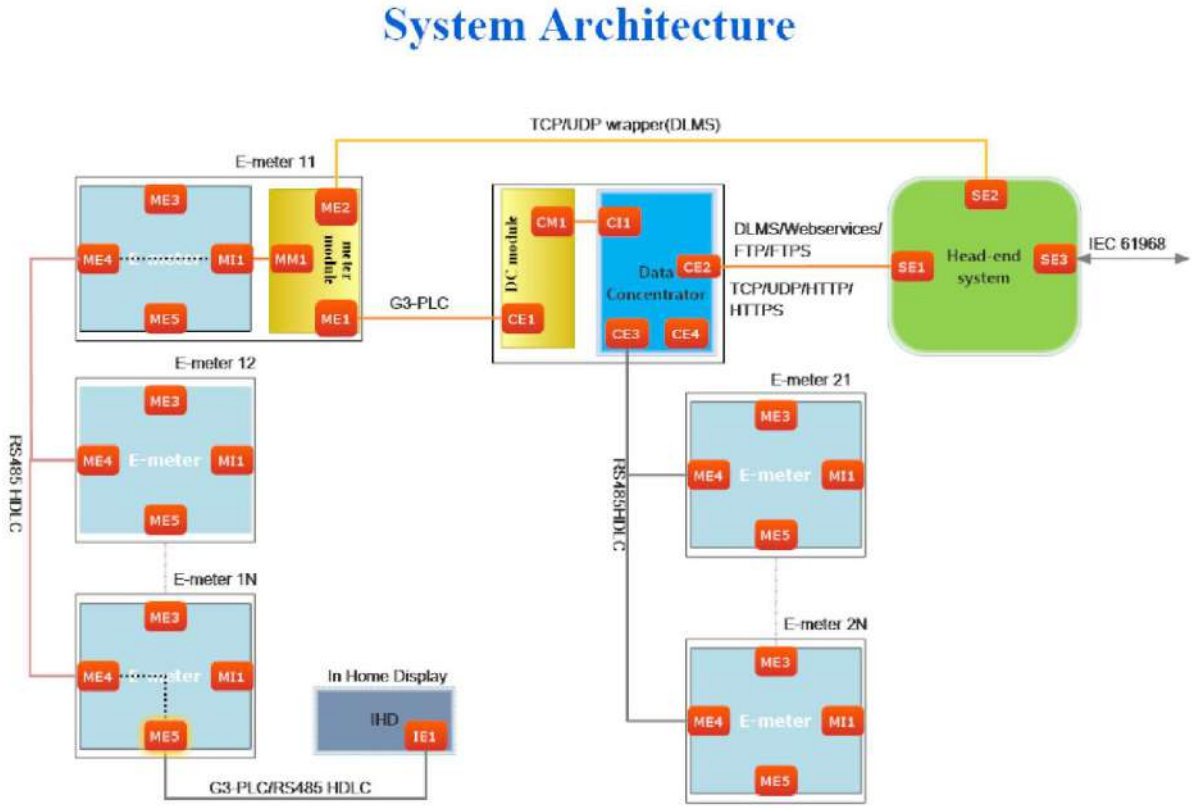
Annexure-B

Unified Meter DataCollection System

Meter Modeling

1. Overview

The system interface architecture of the device is shown in the following figure; the electric energy meter, concentrator and front end system are simplified into several interfaces marked with correspondent code name respectively;



Device functions & Interoperability Interfacing Requirements are composed of 3 parts, including:

Package 0： Addressing System

Package 1： DCU to HES

Package 2： Meter Modeling

2. DLMS Client / Server Application Layer Description

2.1Client / Server Architecture Description

The data exchange between the data acquisition system and the collected equipment, should refer to the communication mode of the client / server AP specified by DLMS. The client AP initiates the access request and the server AP responds, which is a question and answer communication mechanism; and the role of server AP is specified to be assumed by the meter or module.

By means of access management, one server AP can exchange data with one or more client APs at the same time, and a client AP can also access one or more server APs at the same time.

In addition to the normal question and answer mechanism, if there are abnormal events in the server AP, the event can be reported through the prescribed format.

The three-layer architecture of DLMS is shown in the following figure::

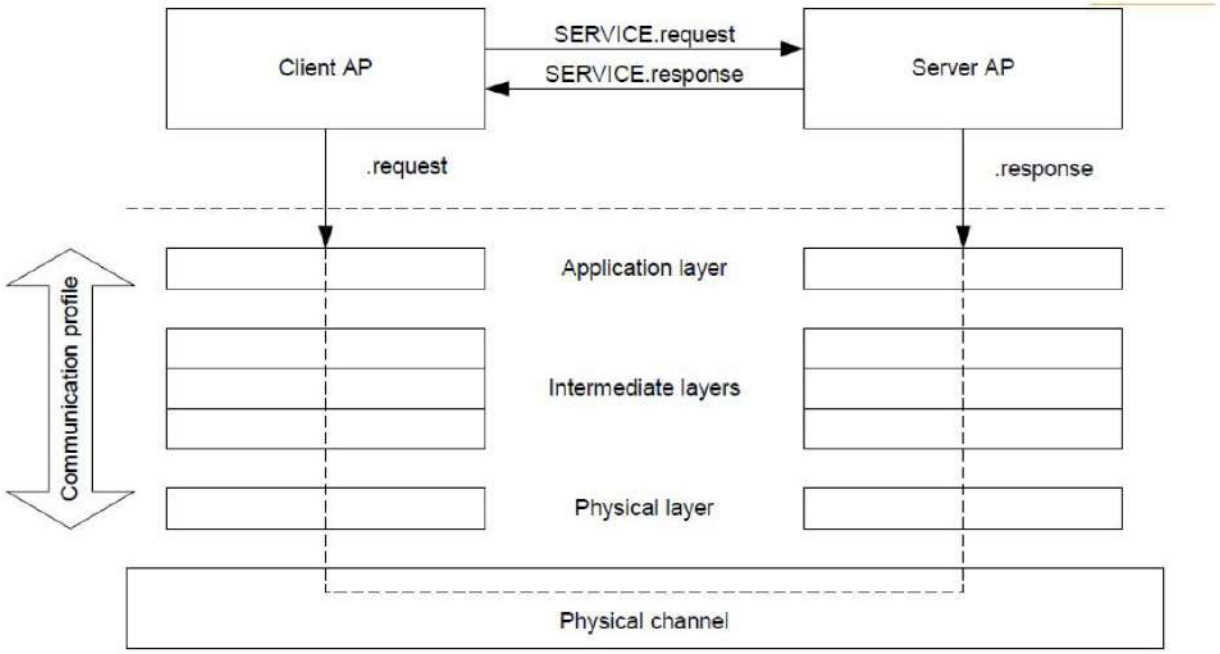


Figure 2 – Client/server relationship and protocols

Level 1: Physical Device Layer
Level 2: Logical Device/Intermediate Layer
Level 3: COSEM Application Layer

The physical device mentioned here may include multiple logical devices (for example, multiple logical devices in one physical device may communicate with different logical device addresses by using the same physical address, so as to be able to access meter or communication Module - two logical devicesindependently); For the energy meter, base meter, the default address of the logical device is:

ServerSAP = 0x01;

Each logical device has an object representing its logical device name, the default logical device name of OBIS:

(1, 0-0:42.0.0.255, Class ID: 1)

Each logical device supports up to 5 types of COSEM clients based on different communication interfaces and project requirements:

· Public Client (SAP: 016)
· Pre-established Client (SAP: 102)
· Management Client (SAP: 001)
· Reading Client (SAP: 002)
· Module Client (SAP: 096)

Public client (SAP: 016); Without any security level protection, it can only read the most basic information about the device, such as serial number, logical device name, SAP support list, etc .; can not program or set.

Pre-link client (SAP: 102); does not need to establish application layer connection; pre-link client link has been there, can not be disconnected, mainly is used to improve the reading efficiency of remote communications. In addition to pre-link and module client, all other clients need to establish application-layer connection through AARQ; support for reading of necessary data items and programming of a small amount of data.

Management Cient (SAP: 001); has administrator rights to operate the electricity meter, can do all the operations.

Read client (SAP: 002); for reading data locally and remotely, able to read almost all data items, but without programming and setting permissions.
Module client (SAP: 096); MI1 interface for data exchange between the meter and the module, pre-link, can read and program some module-related data items; only the MI1 interface supports this client

Client	Functions	Descriptions
Public Client	Read the basic data	Used for local and remote communication; Security level: No security certification Establish the connection via AARQ
Pre-link Client	Read and set some specified data items	Used for remote communication, pre-link, security certification is not required;
Management Client	Own all the administrator permissions for the read/write/action	Used for local and remote communication Security level：HLS（LLS） Establish the connection via AARQ
Read Client	Used for data reading	sed for local and remote communication Security level：HLS（LLS） Establish the connection via AARQ
Module client	User built-in module access the meter quickly	Used for MI1 pre-link, security certification is not required;

2.2PDU maximum size（ServerMaxReceivePduSize）

Electricity Meter, the default maximum negotiated APDU size is 1024 bytes; however, depending on the project and the communication scheme, adjustments may be made

ServerMaxReceivePduSize = 1024 bytes（default）；

3.3 DLMS Application Layer Service

Application layer service by default only supportsLN(logical Name)；

ContextName = LONG_NAMES；

Version No.：

DLMS version= 6；

According to the description of the DLMS standard, the service types supported by the electricity meter are as follows:

Service	Conformance Block Bit
general-protection	1
general-block-transfer	2
block-transfer-with-get	11
block-transfer-with-set	12
multiple-references	14
data-notification	16
Get	19

Set	20
selective-access	21
action	23

For different clients, the default supported services are as follows:

Client	Functions
Public Client	Get block-transfer-with-get
Pre-link Client	get set action data-notification general-block-transfer general-protection
Management Client	get block-transfer-with-get set block-transfer-with-set selective-access multiple-references action general-block-transfer general-protection
Read Client	get block-transfer-with-get selective-access multiple-references general-block-transfer general-protection
Module Client	get block-transfer-with-get set block-transfer-with-set selective-access multiple-references action general-block-transfer general-protection

For each client number with different logical device, there is an application layer connection object, Class15 (Association LN), used to describe the link information such as current link list, link status etc., in addition, you can also conduct HLS Authentication and the key modification through class 15; link object:

“Current Associaon” (0-0:40.0.0.255, Class ID: 15)- urrent link object
“Public Associaon” (0-0:40.0.1.255, Class ID: 15)
“Management Associaon” (0-0:40.0.2.255, Class ID: 15)
“Pre-established Associaon” (0-0:40.0.3.255, Class ID: 15)
“Reading Associaon” (0-0:40.0.4.255, Class ID: 15)
“Module Associaon” (0-0:40.0.5.255, Class ID: 15)

3. Communication Interface

3.1 MI1

The MI1 interface is used for the base energy meter to communicate with the remote communication module MM1 interface in the meter and it is in the server mode. The module types that this interface can support include but are not limited to the following categories:

· Celluar(GPRS/WCDMA/LTE)
· PLC
· RF
· Etherner
· RS485

interface	phy layer	Intermediate layer	COSEM
MI1	SmartII interface SmartIII interface	HDLC: Address length support: 1/2/4 bytes Support UI frame service； The default baud rate： 9600b/s N,8,1	Public Cent Pre-link clientManagement Cient Read client Module client

3.2 ME1

The ME1 interface is used for remote communication and communicates with the CE1 interface of the concentrator, and it is in the server mode. The specific communication architecture is determined by the specific module. It should be noted in the package 1 that the remote channel accesses PDU data of the electricity meter through the ME1 interface,and after processing in the physical layer and link layer in the module, the data will be converted into an HDLC data format and transmitted to the electricity meter through the MM1 interface, the module will not conduct any process to the PDU data. For the details of data conversion, please refer to pakcage 2; the interface of ME1 includes but not limited to the following categories:

· PLC
· RF
· RS485

interface	phy layer	Intermediate layer	COSEM
ME1	PLC RF RS485	6lowPan HDLC Wrapper	Public Cent Pre-link clientManagement Cient Read client

3.3 ME2

The ME2 interface is used for remote communication and communicates with the SE2 interface of the HES,and it is in the server mode. The specific communication architecture is determined by the specific module. It should be noted in the package 1 that the remote channel accesses PDU data of the electricity meter through the ME2 interface,and after processing in the physical layer and link layer in the module, the data will be converted into an HDLC data format and transmitted to the electricity meter through the MM1 interface, the module will not conduct any process to the PDU data. For the details of data conversion, please refer to pakcage 2; the interface of ME2 includes but not limited to the following categories

· Celluar(GPRS/WCDMA/LTE)
· Etherner

interface	phy layer	Intermediate layer	COSEM
ME2	Celluar(GPRS/WCDMA/LTE) Etherner	Wrapper IPv4 IPv6 TCP/UDP	Public Cent Pre-link clientManagement Cient Read client

3.4 ME3

ME3 interface is used for local communication, here mainly refers to the optical interface;and it is in server mode; can be used for local maintenance and programming settings

· optical port

interface	phy layer	Intermediate layer	COSEM
ME3	IEC62056-21 E	HDLC: Address length support: 1/2/4 bytes The default baud rate：9600b/s N,8,1	Public Cent Management Cient Read client

3.5 ME4

The ME4 interface is used for reading other types of utility meters such as water meters, gas meters, heat meters etc.; and it is in the client mode; this interface includes but not limited to the following communication types

· RS485

interface	phy layer	Intermediate layer	App layer
ME3	RS485	1.HDLC: Address length support: 1/2/4 bytes The default baud rate：9600b/s N,8,1	1.COSEM:

3.6 ME5

MEE interface is used for communicating with the indoor display unit; and it is in server mode; this interface includes but not limited to the following types of communications:

·PLC

· RF

· RS485

interface	phy layer	Intermediate layer	COSEM
ME3	PLC RF RS485	HDLC: 6lowPan Wrapper	Public Cent Management Cient Read client

4.6 HDLCPhysical Address

Rules of HDLC device address default are as follows:

1. The value of the last four bits is16-9999; set directly to the meter; For example: address: 037586937378, then RS485 HDLC address: decimal: 7378; hexadecimal number: 0x1CD2

2. The value of the last four bits is 0-15; set to meter after adding 10000; - HDLC prescribes 0-15 address, this area is not allowed to set; For example: address is: 037586930001, then HDLC address is: decimal: 10001; hexadecimal number: 0x2711;

If there is a duplicate address on the same bus during field installation, you can reset it; the range of addresses allowed to be reset is: 0x2720-0x27E7;

3.7 Communication Interface Settings

Each communication channel can be configured with the following objects:

- HDLC Setup3 (23, 0-3:22.0.0.255);
- Depending on the communication module, it is not configured in the meter; Depending on the communication module, it is not configured in the meter;
- HDLC Setup 0 (23, 0-0:22.0.0.255);
- HDLC Setup 4 (23, 0-5:22.0.0.255);

4. DLMS Device Identification

This section is intended to show how devices are named according to the requirements of DLMS / COSEM. Each device corresponds to a unique name (serial number);Provide basis for third-party equipment registration, identification.

Device serial number: "Device ID7" COSEM object (1-0: 0.0.0.255, Class ID: 1); This encoding is unique and by default is a string of 11 digits. Depending on different power company's requirements, this encoding rule will change to supporting a maximum of 14 digits; this code is set by the manufacturer to the device at the factory and is also printed on the device's panel

System ID(system title)： According to different project requirements, each device has a unique 8-byte system identification code, which is fixed to the device before shipment. It is used for mutual authentication and identification between different vendors, different device types, and different device serial numbers in the same system. For details, please see section 5.1.

Logical device name: "COSEM Logical Device Name" COSEM object (0-0: 42.0.0.255, Class ID: 1). As described above, each physical device may correspond to multiple logical devices. The logical device name is a logical device unique identifier, and the specific encoding format is in Section 5.2.

4.1 System ID（SYSTEM TITLE）

Electricity Meter, Power Concentrator, Collector, HES, each device has a unique system ID (8 bytes). According to the definition of DLMS / COSEM, the system ID is as follows

Byte1	Byte2	Byte3	Byte4	Byte5(4-7)	Byte5(0-3)	Byte6	Byte7	Byte8
I	H	M	DT	FT	SNM	SNM	SNM	SNM

Byte1-3： "IHM"ASCII code， vendor identification number registered by the manufacturer in DLMS.

DT: Device type（001-255）

001： 1P BS (Terminal Base) Meter
002: 1P ANSI (Socket Base) Meter
003: 3P Direct Connection Meter
004: 3P CT connection Meter
005: 3P CT/PT connection Meter
006: 1P DIN Meter
007-029: Reserved
030: Data Concentrator
031: Data Collector(RS485 type)
032…39: Reserved
040: Head-end System
041…049: Reserved
050: HHU
051…255: Reserved

FT: Function type. The corresponding position 1 indicates that the device has this function
· bit0=1 Disconnector exists
· bit1=1 Load Management exists
· bit2=1 MulUlity meter is supported
· bit3=0 reserved

SNM: The last 8 digits of the device serial number;Stored in 28-bit SMN in hexadecimal.
For example, if the device serial number of a single-phase BS-mounted electricity meter (DT: 001) is: 37912345678 (0x0BC614E), with load control function and supports the ME4 interface (FT: 0111),Then the device system title is (hexcode).

Byte1	Byte2	Byte3	Byte4	Byte5(4-7)	Byte5(0-3)	Byte6	Byte7	Byte8
49	48	4D	01	7	0	BC	61	4E

4.2 Logical Device Name (0-0:42.0.0.255，Class ID: 1)

The name of the logical device in COSEM is 16 bytes and is defined as follows:

Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
I	H	M	DT	DT	DT	FT	FT

Byte9	Byte10	Byte11	Byte12	Byte13	Byte14	Byte15	Byte16
SNM	SNM	SNM	SNM	SNM	SNM	SNM	SNM

The values of DT, FT and SNM here are the same as the system ID, but the expression format is ASCII; if DT = 0x01, the ASCII code is "001" in the name of the logical device, accounting for 3 bytes.

4.3 Device ID

In addition to the above, the device ID also contains at least the following COSEM objects, depending on the project, the specific definition will be different

Device ID1 0-0:96.1.0.255：Device hardware serial number

Device ID2:0-0:96.1.1.255：Device type identification code

Device ID3:0-0:96.1.2.255：Asset name

Device ID4:0-0:96.1.3.255：GPS info

Device ID5:0-0:96.1.4.255：End user name

Device ID6:0-0:96.1.5.255：Power company certification code

5. Device Automatical Registration

Automatic device registration refers to the process of pushing the device's logical device name and IP address to the HES through the Data-notification service after the device is installed, and the HES reply that the device is successfully registered after receiving.

Automatic registration of the device for the electricity meter, it means through the MT1 interface, using the setting content of the push setup –on installation (0-7.25.9.0.255, class 40) object, send the registration information to the MT1 interface via the service of Data-notification; MI1 interface uses HDLC UI frame to transmit registration information by default; only pre-link client supports Data-notification service by default in the device.

After receiving the data frame, the communication module establishes the data link as soon as possible and reports the information to the upper level device. After the higher level device (concentrator or HES) receives the registration information and confirms it, it replies and set the information of “E_instal”to the meter's user information display (0-0: 96.13.1.255, class 1).

Long press the wheel button on the device for 10 seconds (LCD prompt "install_S"), you can trigger the device to automatically report the register information; At the same time, you can also configure the time of automatic registration and retransmission times, etc. by setting the push setup –on installation.

Push objects of automatical registration by default:

Logical device name：(1, 0-0:42.0.0.255，2);

IPv4 Setup –local IP address (42, 0-0:25.1.0.255, 3);

6. Tariff Management

The electric meter can conduct time-sharing measurementfor active, reactive and apparent electricity consumption.It can define up to eight tariffs, which means that the meter can be divided into eight

different tariff measurement storage unit. The daily maximum can be divided into 24 periods, each time period corresponds to a unique tariff tag.
The tariff conversion of the electricity meter is based on the daily tariff table as the basic unit, the meter can pre-store up to 16 sets of daily tariffs table, support for 50 special holidays by default; And then select corresponding daily tariff tableaccording to the holiday tariff schedule, seasonal tariff schedule, weekly tariff table to achieve period tariff conversion in different holidays, different seasons, different workdays in a week.

During the tariff management process of the electricity Meter, the COSEM need to be involved as follows:
The objects need to be set during the tariff configuration:
· Special Days Table (0-0:11.0.0.255, Class ID: 11);-Current holiday table
· Passive Special Days Table (0-0:11.0.1.255, Class ID:11);-Spare holiday table
· Activity Calendar (0-0:13.0.0.255, Class ID: 20);-Activity calendar tariff schedule
· Default Energy Tariff in Case of Invalid Clock (0-0:96.14.9.255, Class ID: 1);-Time error default energy tariff
· Default Max. Demand Tariff in Case of Invalid Clock (0-0:96.14.10.255, Class ID: 1);- Time error demand tariff
Auxiliary Return (read) Object:
· Clock (0-0:1.0.0.255, Class ID: 8);-Time
· Tariffication Script Table (0-0:10.0.100.255, Class ID:9);-Tariff switch script
· Passive Tariffication Script Table (0-0:10.1.100.255, Class ID:9);-Standby tariff switch script
· Register Activation- Energy(0-0:14.0.1.255, Class ID: 6);- Activity energy register
· Register Activation- Maximum Demand (0-0:14.0.2.255, Class ID: 6);-Activity demand register
· Currently Active Energy Tariff (0-0:96.14.0.255, Class ID: 1);-Current energytariff
· Currently Active Maximum Demand Tariff (0-0:96.14.1.255, Class ID: 1);-Current demand tariff
· Energy Registers (…, Class ID: 3)-Energy register（1.8.0…）
· Maximum Demand Registers (…, Class ID: 4)-Demand register（1.6.0…）

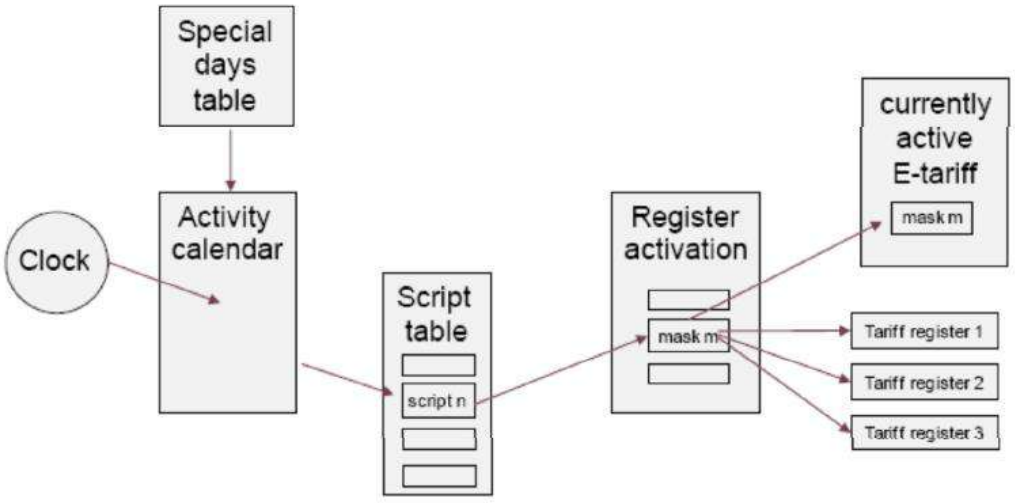


Figure 11: Tariffication Process

Tariff conversion diagram

6.1 Default Tariff

If the device clock fails, then the equipment needs to measure the consumption in accordance with the default tariff set; default tariff settings for the following two COSEM objects:

· Default Energy Tariff in Case of Invalid Clock (0-0:96.14.9.255, Class ID: 1);
· Default Max. Demand Tariff in Case of Invalid Clock (0-0:96.14.10.255, Class ID: 1);

The above two objects can be set up locally or remotely

6.2 Current Tariff

The current running tariff can be obtained by reading the following two objects

· Currently Active Energy Tariff (0-0:96.14.0.255, Class ID: 1);
· Currently Active Maximum Demand Tariff (0-0:96.14.1.255, Class ID: 1);

7. Meter Measurement Data

The acquisition of measurement data of the electricity meter can be carried out through "GET" or "data-notification" service. All the measurement data supported by the electricity meter can be found in the following locations according to different projects.

”Reigster IC 3 Energy”
”Reigster IC 3 parameters”-----“Related parameter of power operation”
”Ext. Reigster IC 4”

The default measurement accuracy is as follows：

Parameter	Unit	Resolution	Scalar
Current	A	0.01	-2
Voltage	V	0.1	-1
Energy (1P, 3P-DC and 3P-CT)	Wh/VARh	1	0
Energy (CT/PT Meter)	Wh/VARh	0.1	-1
Power Factor	-	0.001	-3
Frequency	Hz	0.01	-2
Harmonics Parameters	%	0.1	-1

8. Curve Type

8.1 Curve

The COSEM objects of all the curves can be found in the following positions in the 《OBIS data item details》 according to different projects, which are generally divided into four categories: event record curve, power quality curve, bill curve and ME4 interface curve.

”Profile IC 7 Event records”-Envent record
” Profile IC 7 load profile”-Power quality
” Profile IC 7 billing profile”-Bill
” Profile IC 7 ME4 profile”- Submeter reading

The table contains information about the default capture objects, optional capture objects, capture cycles, and maximum memory for all curves in this project; in addition,
The curve supports two modes of reading in the "selective-access" service, in the time period and the point of point; the curve in the default state:
For the event record curve, it need to capture the event code, and the code definition of the event refer to “Event_list”.

9. Power Quality

Power quality monitoring mainly includes the following items:
·Undervoltage and overvoltage;
·Phase loss (outage);
·Average voltage;

·Sudden drop and raise in voltage (CT/PT Meters);
·THD(CT/PT Meters);
Each harmonic (CT/PT Meters);
It should be noted that, for a certain phase voltage within the same period of time, undervoltage / overvoltage / phase loss / voltage interrupt occurs only in a situation;overvoltage / undervoltage / phase loss/ voltage interruption threshold for determination cannot overlap;if one phase is under undervoltage / overvoltage / phase loss / state when the voltage interruption occurs, so the electricity meter will be considered undervoltage / overvoltage / phase event has been finished.

110%-200%Un	overvoltage
51%-90%Un	undervoltage
11%-50%Un	phase loss
0%-10%Un	voltage interruption

9.1 Undervoltage / Overvoltage

The electricity meter needs to monitor all undervoltage and overvoltage phenomena; if the amplitude of the voltage fluctuation is greater than the set threshold and the duration exceeds the set time, it is judged that the undervoltage / overvoltage event occurs; if the voltage is within Undervoltage / overvoltage condition, returns to normal voltage and maintains above the set threshold, the undervoltage / overvoltage event is over. According to the requirements of the project, over-voltage and under-voltage threshold can be set, setting range, overvoltage 110% -200% of the reference voltage, undervoltage 51% -90% of the reference voltage, unit is %; Set objects are as follows:

- Threshold for Under Voltage (class 3, 1-0:12.31.0.255, 2);
- Threshold for Over Voltage (class 3, 1-0:12.35.0.255, 2);

The default overvoltage judgment time is 15S, the undervoltage judgment time is 60S; the settable range is 1-180S; the start and end events use the same judgment delay, which can be modified by the following two objects:

- Time Threshold for Over Voltage (3, 1-0:12.44.0.255, 2);
- Time Threshold for Under Voltage (3, 1-0:12.43.0.255, 2);

If an overvoltage or undervoltage event has occurred; then the start and end event record needs to be recorded in the below curve:

- Power Quality Event Log (7, 0-0:99.98.4.255);（Specific capture object, please see “OBIS data item details”）

In addition, the electricity meter also records the total number of overvoltage / undervoltage events, the last time they occurred, the percentage of the voltage at the last trigger event relative to the reference voltage, and is stored in the following objects:

- Number of Under Voltage in Phase L1 (1, 1-0:32.32.0.255);
- Number of Under Voltage in Phase L2 (1, 1-0:52.32.0.255);
- Number of Under Voltage in Phase L3 (1, 1-0:72.32.0.255);
- Dura on of Last Under Voltage in Phase L1 (3, 1-0:32.33.0.255);
- Dura on of Last Under Voltage in Phase L2 (3, 1-0:52.33.0.255);
- Dura on of Last Under Voltage in Phase L3 (3, 1-0:72.33.0.255);
- Magnitude of Last Under Voltage in Phase L1 (3, 1-0:32.34.0.255);
- Magnitude of Last Under Voltage in Phase L2 (3, 1-0:52.34.0.255);
- Magnitude of Last Under Voltage in Phase L3 (3, 1-0:72.34.0.255);
- Number of Over Voltage in Phase L1 (1, 1-0:32.36.0.255);
- Number of Over Voltage in Phase L2 (1, 1-0:52.36.0.255);
- Number of Over Voltage in Phase L3 (1, 1-0:72.36.0.255);
- Dura on of LastOver Voltage in Phase L1 (3, 1-0:32.37.0.255);

- Dura on of LastOver Voltage in Phase L2 (3, 1-0:52.37.0.255);
- Duration of LastOver Voltage in Phase L3 (3, 1-0:72.37.0.255);
- Magnitude of Last Over Voltage in Phase L1 (3, 1-0:32.38.0.255);
- Magnitude of Last Over Voltage in Phase L2 (3, 1-0:52.38.0.255);
- Magnitude of Last Over Voltage in Phase L3 (3, 1-0:72.38.0.255);

9.2 Phase loss

If the meter detects that the grid voltage is lower than the set phase loss threshold and the duration is greater than the set threshold, then the phase loss event isconsidered as occurring. If the grid voltage is restored to normal and continues for longer than the set threshold time, it is considered that the phase loss event has ended. The start and end events of phase loss in any phase of L1, L2, and L3 need to be recorded in the lower curve

- Power Quality Event Log (7, 0-0:99.98.4.255);

Configuration objects for phase-missing event thresholds:

- Threshold for Voltage Cut(Class ID: 3, 1-0:12.39.0.255,2);Default value： 50% Vref;setting range 11%-50%;
- Time Threshold for Voltage Cut (Class ID: 3, 1-0:12.45.0.255,2);Default value： 30 seconds, setting range 1-180 seconds;

9.3 Average Voltage

The electricity meter can monitor the average value of each phase voltage over a period of time, the calculation cycle can be set in the range of 1-60 minutes with the COSEM object, the default calculation period is 10 minutes, the calculation must be at least 1 cycles before an average, then all phases power down, it does not calculate; Related COSEM objects are as follows;

- MeasurementPeriod 3 for Instantaneous Value (3, 1-0:0.8.2.255, 2)-Calculation period， default 10min，1-60分钟；
- Average voltage L1 (Class ID: 3, 1-0:32.24.0.255);-L1average voltage
- Average voltage L2 (Class ID: 3, 1-0:52.24.0.255);-L2 average voltage
- Average voltage L3 (Class ID: 3, 1-0:72.24.0.255);-L3 average voltage

According to EN 50160's weekly average voltage pass rate requirements, and based on the average voltage calculated above, the electricity meter carry out the statistics of pass rate for each of the above calculated average voltage within the past seven days; If in the past 7 days (the default calculation period: start from 0:00 every Sunday), the average voltage greater than 5% exceeds the range of +-10% Uref, the it needs to record the voltage and quality difference events in the following curves.

- Power Quality Event Log (7, 0-0:99.98.4.255);

The event code corresponding to the voltage quality difference is as follows:

- Event Code 92: Bad Voltage Quality L1
- Event Code 93: Bad Voltage Quality L2
- Event Code 94: Bad Voltage Quality L3

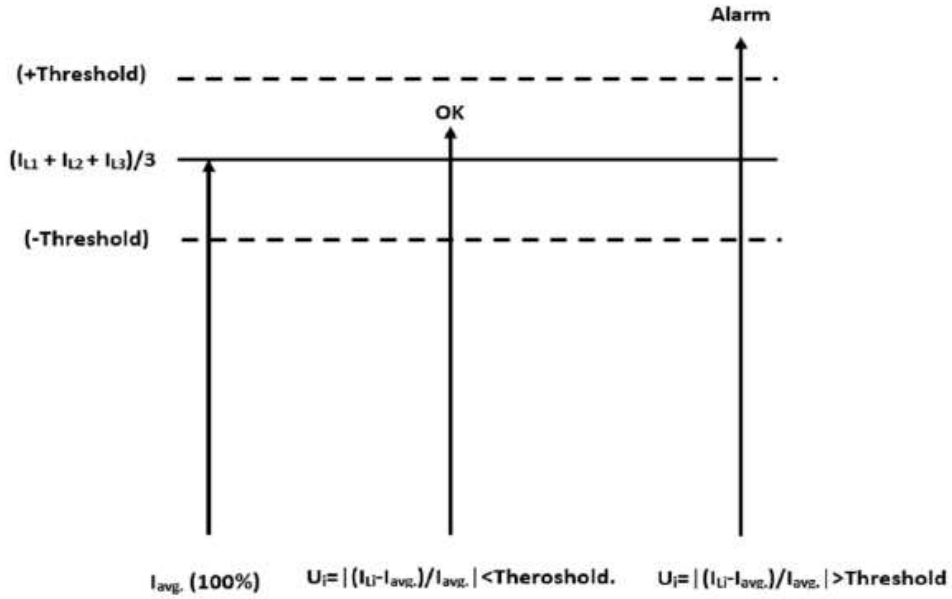
9.4 Harmonic/THD Measurement（3P CT/PT Meters）

For the CT / PT meter, the watt-hour meter has the function of measuring and recording the 31st harmonic component. At the same time, the THD harmonic content of each phase voltage and current can be recorded separately. The corresponding COSEM object can be found in the OBIS list class3

9.5 Load Imbalance (Three-phase Meter)

If the current of each phase is greater than the set unbalanced load judgment current threshold and the difference between the average current value of any one phase and the average value of the three-phase currents is greater than the set judgment threshold, then it is considered that the load imbalance occurs, In this state, the UBL of the status word Profile Status 1 (1, 0-0: 96.10.1.255) in all the curves will be set; at the same time, a load imbalance event will also be generated (specific event code referring to event code List) and recorded in the following curve

Power Quality Event Log (7, 0-0:99.98.4.255);



As shown in the above figure, Ili is the average current value of the previous calculation period of Li phase;the calculation period of this average current is determined by the capture period set in the average curve (class ID: 7,1-0: 99.133.0.255) Default value is 15 minutes). When a new average value is calculated, the electricity meter will judge whether it is in the unbalanced load power supply mode according to the above rules, and confirm whether to report the event according to the setting.
It should be noted that the calculation of the average must be full cycle; related settings COSEM object:
UnbalanceLoad Detection (1, 1-0:96.98.15.255)
The data structure of this object is as follows:
{
long_unsigned minimum current that activate unbalance task inmeter (default 1A)
long_unsigned Unbalance threshold (default 50%)
}
Current object used to determine the imbalance event:
Last Average Value of Current L1 (3,1-0:31.25.0.255,2)
Last Average Value of Current L2(3,1-0:51.25.0.255,2)
Last Average Value of Current L3(3,1-0:71.25.0.255,2)

10. Maximum Demand

Demand: the average power within the specified time.
Period of demand: a continuous equal interval of time for measuring the average power.
Maximum Demand: The maximum value of demand recorded during a specified period of time.

10.1 The Maximum Demand Calculation Instructions

- In the agreed time interval (usually a month or a meter reading settlement cycle), measure the forward active, reverse active, combined reactive 1, combined reactive 2, forward apparent, reverse apparent maximum demand;Sub-period forward active, reverse active, combined reactive 1, combined reactive 2, forward apparent, reverse apparent maximum demand and the date and time of maximum demand occurs.
- The maximum demand measurement can be programmed to choose slip mode or interval mode.
- Demand cycle and slip time can be set.
- Demand cycle can be selected in 5,10,15,30,60 min.

Slip demand cycle slip time can be selected in 1, 2, 3, 5min.

The demand cycle should be an integral multiple of 5 for the slip time. Factory default: demand cycle 15min, slip time 3min.

- The maximum demand measurement is continuous

The measurement of the maximum demand for each tariff period is carriedout within the complete measurement period of the corresponding tariff period.

When power-on, reset, clock adjustment, period conversion, demand cycle changes, power flow direction conversion, etc. occurs on the voltage line, the meter should start from the current moment, masure the demand according to demand cycle.

When the first demand cycle is completed, start the maximum demand record according to the slip interval.

In the incomplete demand cycle, do not record the maximum demand

10.2 Maximum Demand Settlement

- Automatic settlement: At the appointed daily / monthly settlement day at zero hour, the meter automatically stores the maximum demand and the date and time of the maximum demand within the settlement day / month; and makes the current maximum demand register unit Reset, restart maximum demand calculation of the new meter reading settlement date / month.
- Manual settlement: When the meter parameters are set or the operating conditions are changed, the maximum demand for the current meter settlement date / month needs to be recalculated. You can calculate the maximum demand with the sealing device on the panel Key to make the energy meter store the maximum demand and the date and time of the maximum demand in the settlement cycle; and reset the current maximum demand register unit, and re-calculate the maximum demand of the new meter reading settlement day / month .

Note: Demand automatic settlement and manual settlement mode can be set, the two settlement methods can not be valid at the same time.

10.3 Maximum Demand Clearance

- The electric energy meter can achieve maximum clearance through the infrared communication interface. The maximum demand clearance has a programming switch and password limit.

- In automatic settlement mode, reset the maximum demand register unit in this settlement day / month by using the maximum demand settlement button with sealing device on the panel.

10.4 Maximum Demand Storage

Meter can store the maximum demand each month and the maximum demand occurred date and time of the past 13 months,

		Monthly Data	Daily Data
Forward Active Maximum Demand	Total, per phase, per rate	13 months	124 days

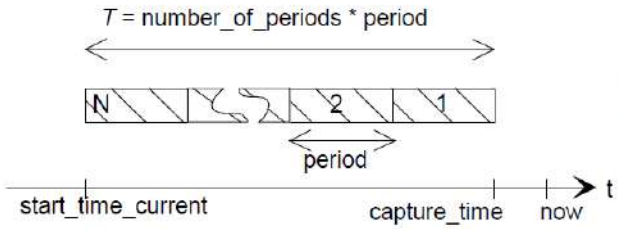
Reverse Active Maximum Demand	Total, per phase, per rate	13 months	124 days
Reactive Maximum Demand Combination 1	Total, per phase, per rate	13 months	124 days
Reactive Maximum Demand Combination 2	Total, per phase, per rate	13 months	124 days
Forward Apparent Maximum Demand	Total, per phase, per rate	13 months	124 days
ReverseApparent Maximum Demand	Total, per phase, per rate	13 months	124 days

10.4 Maximum Demand Calculation Object （Class ID5）

For the above calculation, the class structure of Class ID5 is implemented. The structure contains the current average value, the previous average value, the previous average value generation time, the current period starting time, and the calculation period as shown in the following figure:

Demand register		0...n	class_id = 5, version = 0			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. current_average_value	(dyn.)	CHOICE			0	x + 0x08
3. last_average_value	(dyn.)	CHOICE			0	x + 0x10
4. scaler_unit	(static)	scal_unit_type				x + 0x18
5. status	(dyn.)	CHOICE				x + 0x20
6. capture_time	(dyn.)	octet-string				x + 0x28
7. start_time_current	(dyn.)	octet-string				x + 0x30
8. period	(static)	double-long-unsigned	1			x + 0x38
9. number_of_periods	(static)	long-unsigned	1		1	x + 0x40
Specific methods		***m/o***				
1. reset (data)		o				x + 0x48
2. next_period (data)		o				x + 0x50

The calculation cycle =period*number_of_periods demand for period and number_of_periods, can be individually set, when number_of_periods>1, as the row difference calculation method, when the number_of_periods =1 as interval calculation; number_of_periods set the range of 1-60; period can choose between 300, 600, 900, 1800, 3600 seconds; as shown below:



T is the time interval used for the calculation of the current_value of a sliding demand register

The demand object has two values: the current average value and the last average value.
The current average value refers to the calculated value from the beginning time of the current demand calculation cycle to the current time cumulative energy /T (period*number_of_periods).
The last average refers to the value calculated from the start time of the current demand calculation cycle and the energy / T (period * number_of_periods) accumulated at the end of the previous period;

Note: the last average value is updated after every period. When the last average value is greater than the value in the corresponding object in classID 4, the maximum demand in classID 4 will be updated.

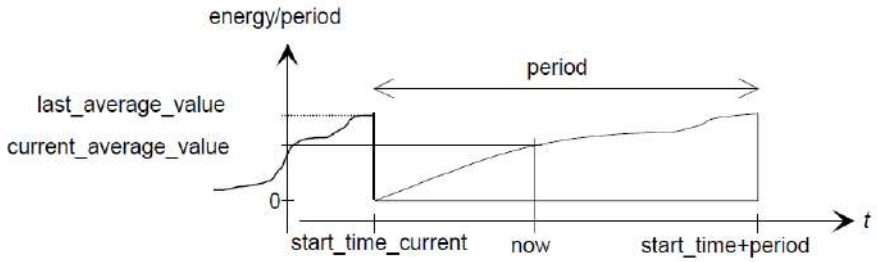
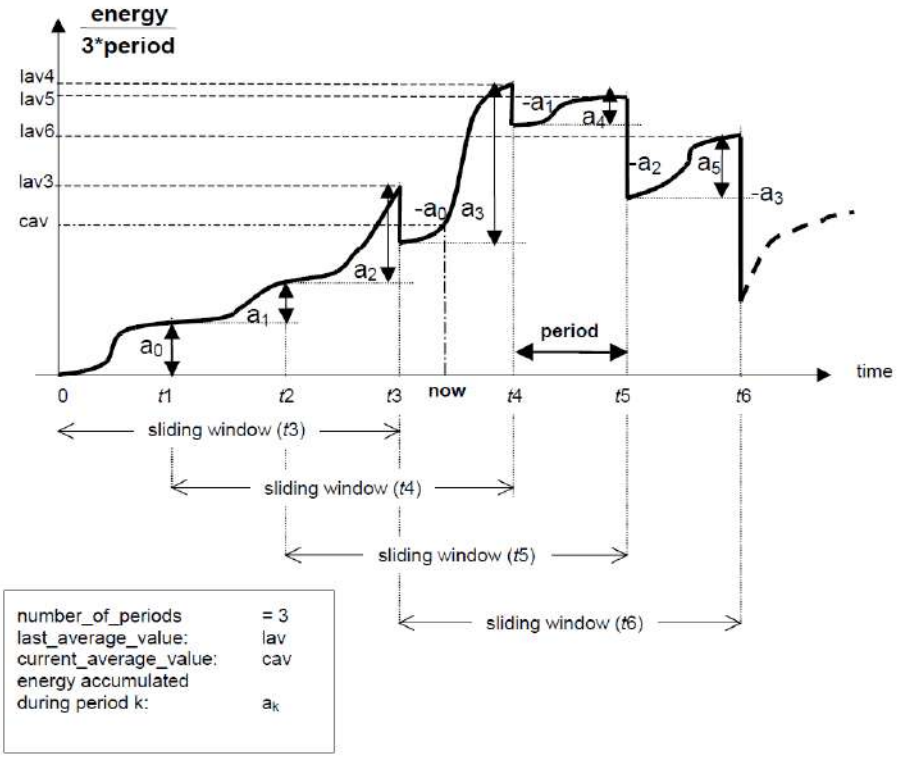


Figure 9 – The attributes in the case of block demand



11. Voltage Interruption

After the meter has been powered up for a minimum of 5S (configurable 1-120s, minimum 5S by default) and the voltage supply at any phase of the meter is below the minimum voltage supply (0.1 Uref by IEEE) A voltage interruption will occur.
When the voltage interruption time is greater than or equal to the setting of "Time Threshold for Long Power Failure" (0-0: 96.7.20.255, Class ID: 3) (default 3 minutes, 1-30 minutes configurable range).It is considered as Long time voltage interrupt event;
If the voltage interruption time is less than the long power failure , it is considered that a short voltage interruption event. If the voltage interruption event occurs on all the voltages of the phases, the power down event occurs.

- Short Power Failure/Interruption (Simply “Power Failure”);
- Long Power Failure/Interruption;
- Power Down;

Short Power Failure event will not be recorded, but the number occurance is recorded in the following objects:

- Number of Short Power Failures in All Phases (0-0:96.7.0.255,Class ID: 1);
- Number of Short Power Failures in L1 (0-0:96.7.1.255,Class ID: 1);
- Number of Short Power Failures in L2 (0-0:96.7.2.255,Class ID: 1);
- Number of Short Power Failures in L3 (0-0:96.7.3.255,Class ID: 1);
- Number of Short Power Failure in Any Phases (0-0:96.7.21.255,Class ID: 1);

After a Long Power Failure event occurs at any phase or all phase voltages, at the end of the event; the end time of the event, the type of event, and the duration are recorded in the "Power Failure" event curve;
At the same time, the number of occurrence of such events and the duration of the last long voltage interruption shall be separately recorded in the data items of class 1 and class 3 below;

- Power Failure event log (1-0:99.97.0.255,Class ID: 7)
- Number of Long Power Failures in All Phases (0-0:96.7.5.255,Class ID: 1);
- Number of Long Power Failures in Phase L1 (0-0:96.7.6.255,Class ID: 1);
- Number of Long Power Failure in Phase L2 (0-0:96.7.7.255,Class ID: 1);
- Number of Long Power Failure in Phase L3 (0-0:96.7.8.255,Class ID: 1);
- Number of Long Power Failure in Any Phase (0-0:96.7.9.255,Class ID: 1);
- Duration of Last Long Power Failure in All Phases (0-0:96.7.15.255,Class ID: 3);
- Duration of Last Long Power Failure in Phase L1 (0-0:96.7.16.255,Class ID: 3);
- Duration of Last Long Power Failure in Phase L2 (0-0:96.7.17.255,Class ID: 3);
- Duration of Last Long Power Failure in Phase L3 (0-0:96.7.18.255,Class ID: 3);
- Duration of Last Long Power Failure in Any Phase (0-0:96.7.19.255,Class ID: 3);

The power up/power down events are recorded in "standard event log".

- standard event log (1-0:99.98.0.255,Class ID: 7)

Note：
1: overvoltage (110%-200%Uref), undervoltage (51-90%Uref), phase 10-50% (Uref) and voltage interruption (<=10% Uref) are mutually exclusive events which means only one event can happens at the same time. If voltage interrupt occurs at the time of undervoltage,the undervoltage event will end

12. Relay Control

The logic of the relay action in the electric energy meter is controlled by the following three ways:

- Remote Control;
- Manual;
- Locally (as internal process);

Remote control refers to the control of the relay through communication interface; manual control refers to control of relay by physical buttons on the meter; Local control refers to control of relay by judgment logic in the electric energy meter program, such as overload, overvoltage events which will activate the relay control;
According to the definition of COSEM classID70, the relay control is divided into three states:

- Disconnected;
- Ready for Reconnection;
- Connected;

The conversion of the three states is shown in the following figure:

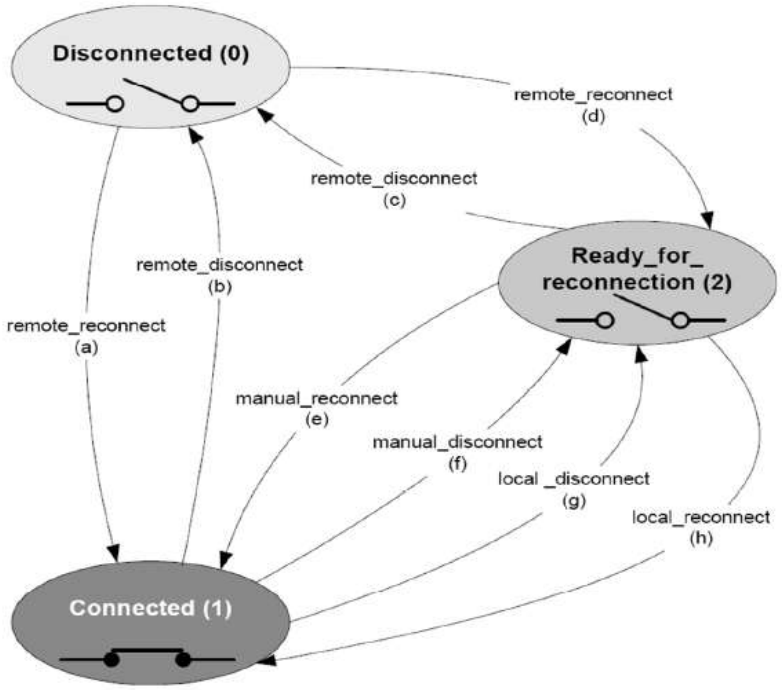


Figure 23: Disconnect Control, State Diagram and Transitions

As shown above, the three states of relay can be converted through 8 ways: "a" -- "H"; By configuring different control mode (70, 0-0:96.3.10.255, 4), the state switching method is selected, and the default state is control_mode = 6.

control _mode	Disconnection				Reconnection			
	Remote		Manual	Local	Remote		Manual	Local
enum:	(b)	(c)	(f)	(g)	(a)	(d)	(e)	(h)
(0)	–	–	–	–	–	–	–	–
(1)	x	x	x	x	–	x	x	–
(2)	x	x	x	x	x	–	x	–
(3)	x	x	–	x	–	x	x	–
(4)	x	x	–	x	x	–	x	–
(5)	x	x	x	x	–	x	x	x
(6)	x	x	–	x	–	x	x	X
NOTE 3 In Mode (0) the disconnect control object is always in 'connected' state.								
NOTE 4 Local disconnection is always possible unless the corresponding trigger is inhibited.								

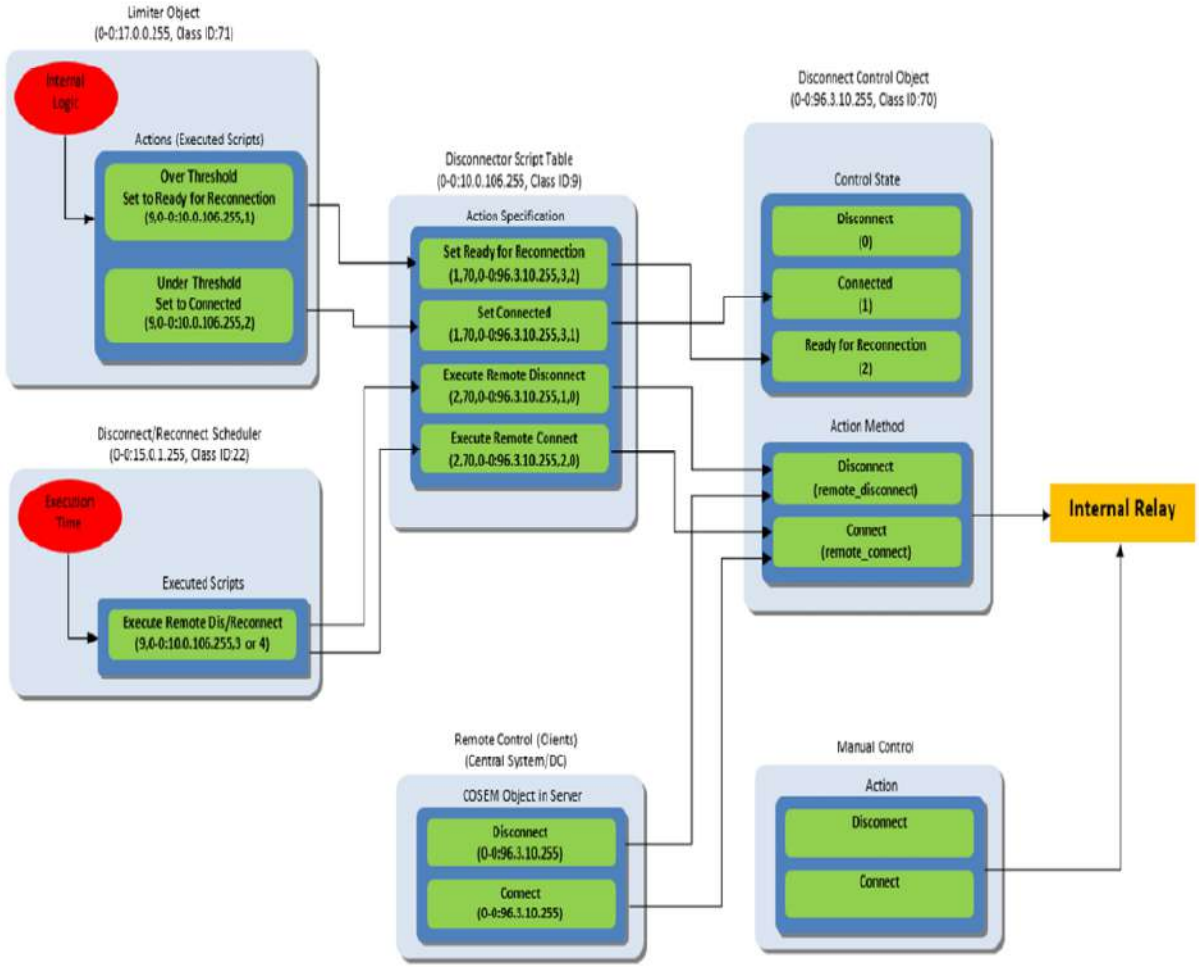
Example: electric energy meter in mode 6, the current relay is disconnected.
The remote system issues a remote reconnect command to the electric energy meter.
Mode 6 supports the path (d), the meter receives the command, the relay status will switch to ready_for_reconnection.
At this point if you want the relay closed (converted to connected) state, you need to operate through the path (e); the specific operation is done on physical keys;

12.1Relay Control Related COSEM Objects：

The list of objects related to the relay control function is as follows:

- Disconnect Control (0-0:96.3.10.255, Class ID: 70)
- Disconnect/Reconnect Control Scheduler (0-0:15.0.1.255, Class ID: 22);
- Disconnector Script Table (0-0:10.0.106.255, Class ID: 9);
- Limiter (0-0:17.0.0.255, Class ID: 71);
- Reclosing Configura on (0-0:96.98.10.255, Class ID: 1);
- Supervision monitor L1 (21, 1-0:31.4.0.255);
- Supervision monitor L2 (21, 1-0:51.4.0.255);
- Supervision monitor L3 (21, 1-0:71.4.0.255);
- Limiter Ac vate Scheduler (22, 0-0:96.98.11.255);
- Limiter Threshold Scheduler (22, 0-0:96.98.12.255);
- Limiter Script Table (9, 0-0:96.98.13.255);

The following diagram is the logic block diagram of the relay control:



13. Billing

The electric energy meter can store the historical billing data history for 13 months. The bill function is described by class ID:22 calling script MDI Reset/End of Billing Period (class id:9, 0-0:10.0.1.255); the related OBIS code of the bill function is as follows:

- MDI Reset/End of Billing Period (class ID:9, 0-0:10.0.1.255);

- End of Billing Period 1 Scheduler (class ID:22, 0-0:15.0.0.255);
- Data of Billing Period 1 (monthly) (class ID:7, 0-0:98.1.0.255);
- Disable/Enable Manual Demand Reset (class ID:1,0-0:96.98.15.255);

The date of the bill is set through End of Billing Period 1 Scheduler (class ID:22, 0-0:15.0.0.255,4). By default, it is the 00:00:00 ("00000000", "FFFFFF01FF") on the first day of the month and the "FF" means not to specify. The action of settlement is as follows:
Maximumdemand in reseting month {class ID:4, 1-0:X.6.Y.255} (X is the power type, Y is the rate numbered) is all the data items in the register;
Freeze all required reading power data registers, store them in Data of Billing Period 1 (monthly) (class ID:7, 0-0:98.1.0.255);
Freeze all types of power data registers and store them in their monthly billing curves.
For the curve Data of Billing Period 1 (monthly) (class ID:7, 0-0:98.1.0.255,3), the capture objects can be configured according to project requirements, which is used for remote reading of monthly historical data. The list of capture objects is as follows:
The bill curve Data of Billing Period 1 (monthly) (class ID:7, 0-0:98.1.0.255,3) supports 50 capture objects, and storage of 13 historical bills can be set up according to different projects.

Different types of electric energy exist in the following curves respectively.
For specific projects, this kind of curve capture object cannot be set
Data of Billing of Period 1 (monthly) 1 (class, ID:7, 0-0:98.1.1.255) - all energy data
Data of Billing of Period 1 (monthly) 2 (class, ID:7, 0-0:98.1.2.255) - all damand data;

14. Meter Clock

Time by "Clock" COSEM object (0-0:1.0.0.255, Class ID 8), and time related items are as follows:
· Clock - time {class ID:8,0-0:1.0.0.255, 2}, - read and set the current time
· Clock -UTC{class ID:8,0-0:1.0.0.255, 3}, - minutes from UTC
· Clock - time state {class ID:8,0-0:1.0.0.255, 4}, - read-only
· Clock - summer time starting time {class ID:8,0-0:1.0.0.255, 5}
· Clock - summer time end time {class ID:8,0-0:1.0.0.255, 6},
· Clock - daylight saving time change time {class ID:8,0-0:1.0.0.255, 7}, - minutes of time change for daylight saving time conversion
· Clock - start summer time {class ID:8,0-0:1.0.0.255, 8}, - 1, activate summer time switching, 0, prohibit summer time switching
·Clock - time benchmark {class ID:8,0-0:1.0.0.255, 9}, - read-only
Time state corresponding bits:
bit 0 (LSB): invalid a value,
bit 1: doubtful b value,
bit 2: different clock base c,
bit 3: invalid clock status d,
bit 4: reserved,
bit 5: reserved,
bit 6: reserved,
bit 7 (MSB): daylight saving active
Time benchmarks:
enum: (0) not defined,
(1) internal crystal,
(2) mains frequency 50 Hz,

(3) mains frequency 60Hz,
(4) GPS (global positioning system),
(5) radio controlled

14.1 Time CalibrationEvent Record

When the time difference before and after calibration is greater than the value in the COSEM object "Clock Time Shi Limit" (Class ID 3,1-0: 0.9.11.255, 2), a calibration event record is generated and stored in the curve Standard Event log (0-0: 99.98.0.255, Class ID 7),
Two types of events need to be captured for the calibration event: 1.Clock Adjusted 1 (Event Code: 4) records the time before calibration; 2.Clock Adjusted 2 : 5) record the new time after calibration
If the time difference before and after proofreading is less than "Clock Time Shi Limit" (Class ID 3,1-0: 0.9.11.255, 2), no event record is generated; (Class ID 3,1-0 : 0.9.11.255, 2) The values can be configured in "Clock Time Shi Limit";

14.2 Display Item

For the LCD display item, the OBIS encoding of the time is as follows (this OBIS is used only for display):

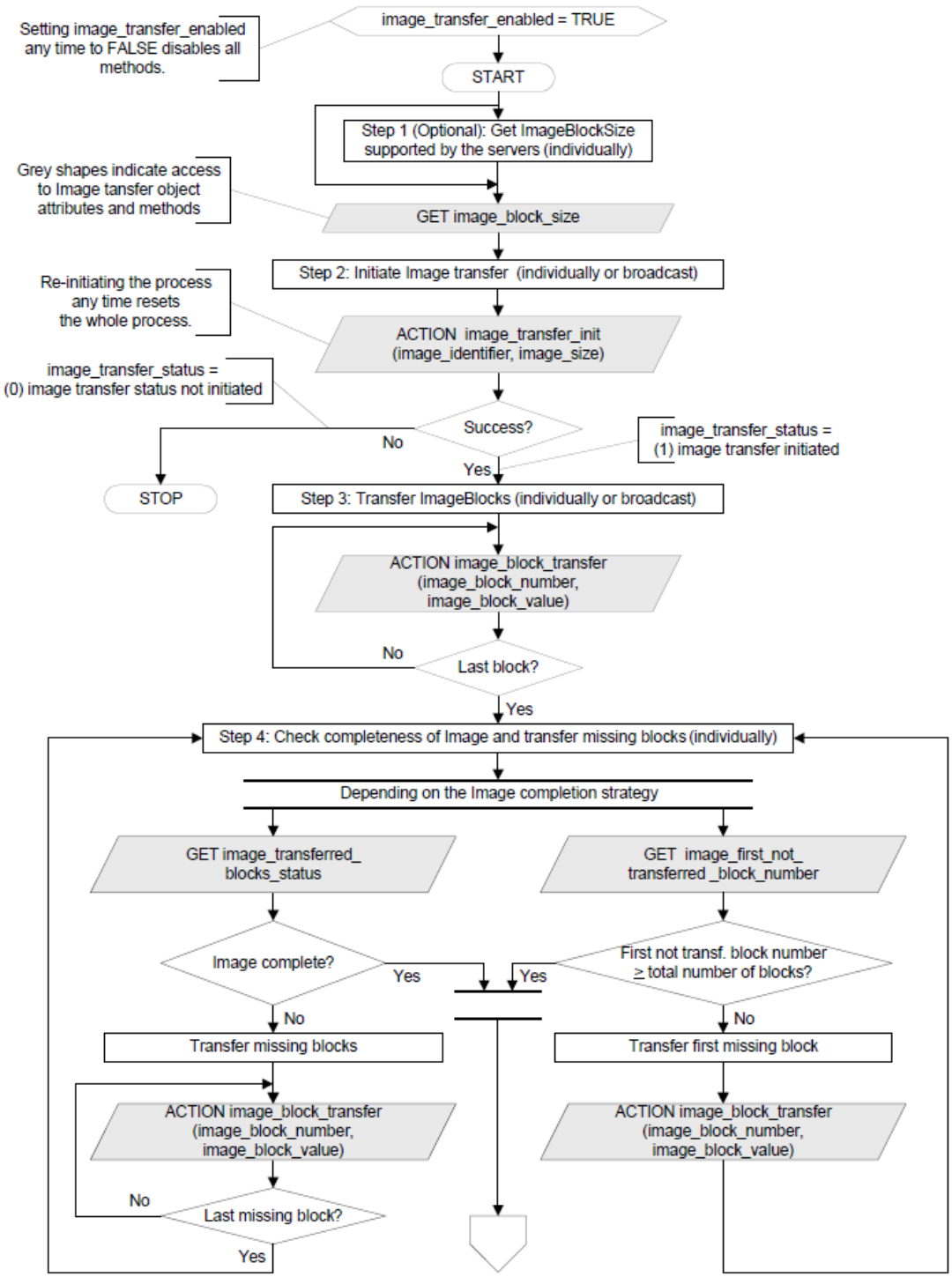
- Local Time (1-0:0.9.1.255, Class ID 1);
- Local Date (1-0:0.9.2.255, Class ID 1);

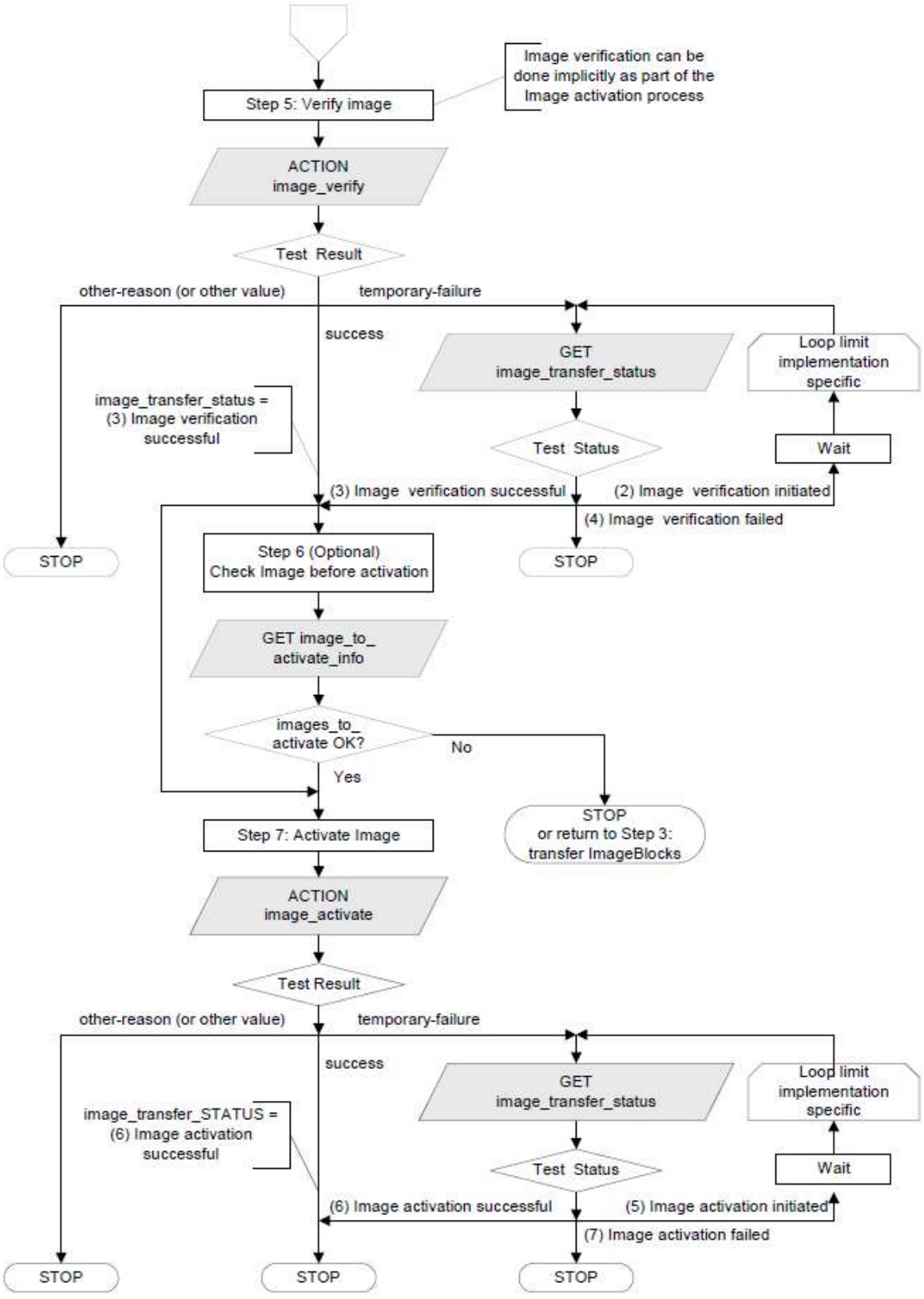
15 . Firmware Update

The electric energy meter and its internal modules can run upgrade firmware through the communication interface; through the class IC DLMS/COSEM 18 image transfer service; electric energy meter / equipment can store two versions of firmwareat the same time, support breakpoint and power off resume in the process of upgrading the firmware. Upgrade related objects are as follows:

- Image transfer (0-0:44.0.0.255, Class ID: 18);
- Image Transfer Activation Scheduler (0-0:15.0.2.255, Class ID: 22);
- Image Activation Script (0-0:10.0.107.255, Class ID: 9);
- Active Firmware Identifier (Metrology Relevant Firmware)(1-0:0.2.0.255, Class ID: 1);- version number
- Active Firmware Identifier 1 (Meter Application Relevant Firmware) (1-1:0.2.0.255, Class 1); - version number
- Active Firmware Identifier 2 (GPRS Communica on Module Firmware) (1-2:0.2.0.255, Class1); - version number
- Active Firmware Signature (1-0:0.2.8.255, Class ID: 1); - verification and (digital signature)
- Active Firmware Signature 1 (1-1:0.2.8.255, Class ID: 1); - verification and (digital signature)
- Active Firmware Signature 2 (1-2:0.2.8.255, Class ID: 1); - verification and (digital signature)

The upgrade process is as follows:
Step1:Read image block size A（class 18 0.0.44.0.0.255 attribute 2）;
Step2:Initiate image transfer(class 18 0.0.44.0.0.255 action 1);
Step3: transfer image block in size A from the first to the last (class 18 0.0.44.0.0.255 action 2)
Step4:Read image transferred block status（class 18 0.0.44.0.0.255 attribute 3）; if finished continue, otherwise transfer the missing blocks;
Step5:verify image (class 18 0.0.44.0.0.255 action 3);
Step6:set the new image activate time（class 22 0-0:15.0.2.255 attribute 4）or Activate Image Immediately(class 18 0.0.44.0.0.255 action 4)
Step7:end。
The flow chart is as follows:


Setting image_transfer_enabled any time to FALSE disables all methods.
image_transfer_enabled = TRUE
START
Step 1 (Optional): Get ImageBlockSize supported by the servers (individually)
Grey shapes indicate access to Image tansfer object attributes and methods
GET image_block_size
Step 2: Initiate Image transfer (individually or broadcast)
Re-initiating the process any time resets the whole process.
ACTION image_transfer_init (image_identifier, image_size)
image_transfer_status = (0) image transfer status not initiated
Success?
No
image_transfer_status = (1) image transfer initiated
Yes
STOP
Step 3: Transfer ImageBlocks (individually or broadcast)
ACTION image_block_transfer (image_block_number, image_block_value)
No
Last block?
Yes
Step 4: Check completeness of Image and transfer missing blocks (individually)
Depending on the Image completion strategy
GET image_transferred_ blocks_status
GET image_first_not_ transferred _block_number
Image complete?
Yes
First not transf. block number ≥ total number of blocks?
Yes
No
No
Transfer missing blocks
Transfer first missing block
ACTION image_block_transfer (image_block_number, image_block_value)
ACTION image_block_transfer (image_block_number, image_block_value)
No
Last missing block?
Yes



15.1 Firmware Upgrade Event Record

There are the following types of events in the firmware upgrade process:
Firmware Ready for Activationevent code: 17
FirmwareVerification Failed, event code: 51
Firmware Activated, event code: 18

Firmware Ready for Activation event code: 17 event occurs when the upgrade package transfer is completed and verified.
If the verification fails then produce Firmware Verification Failed, event code: 51 event occurs.
When the new firmware is activated, a Firmware Activated, event code: 18 event is generated.
These three events are stored in Standard Event (log curve 0-0:99.98.0.255, Class, ID) in 7;

16. Event Management

Event log mainly records the time of occurrence of events, the types of events (event code), and the information when event occurred (depending on the type of events). There are mainly two kinds of information:
Normal event
Alarm event
Normal events are generally stored only through the event curve, and are not actively reported and alerted. Alarm events can be configured by configuring Alarm Filter 1 (1, 0-0: 97.98.10.255) / Alarm Filter 2 (1, 0-0: 97.98.11.255) to choose whether to take the initiative to report on specific events; The following is the event management related objects:

- Error register 1 (1,0-0:97.97.0.255);--Bit definition is the same as Alarm Register 1
- Alarm Register 1 (1, 0-0:97.98.0.255);
- Alarm Register 2 (1, 0-0:97.98.1.255);
- Alarm Descriptor 1 (1, 0-0:97.98.20.255);
- Alarm Descriptor 2 (1, 0-0:97.98.21.255);
- Alarm Filter 1 (1, 0-0:97.98.10.255);
- Alarm Filter 2 (1, 0-0:97.98.11.255);
- Alarm Monitor 1 (21, 0-0:16.1.0.255);
- Alarm Monitor 2 (21, 0-0:16.1.0.255);
- Push Script Table(9, 0-0:10.0.108.255);
- Push Setup-Alarm (40, 0-4:25.9.0.255);
- Standard Events Log (7, 0-0:99.98.0.255);
- Fraud Detection Events Log (7, 0-0:99.98.1.255);
- Disconnect Control Events Log (7, 0-0:99.98.2.255);
- Power Quality Event Log (7, 0-0:99.98.4.255);
- Communication Events Log (7, 0-0:99.98.5.255);
- Power Failure Events Log (7, 1-0:99.97.0.255);

16.1 Event Type And Curve Object

The default events are divided into the following seven categories:

- Standard Events Log (7, 0-0:99.98.0.255);
- Fraud Detection Events Log (7, 0-0:99.98.1.255);
- Disconnect Control Events Log (7, 0-0:99.98.2.255);
- Power Quality Log (7, 0-0:99.98.4.255);
- Communica on Events Log (7, 0-0:99.98.5.255);
- Power Failure Event Log (7, 1-0:99.97.0.255);

Each event corresponds to different capture object.
The capture objects of event log are generally not configurable.
The storage method of event log curve is FIFO first in first out mode, and the maximum number of records depends on the project.

16.2 Events Reading

The events reading can be done selectively with class IC7 which supports reading of event log by point segment or time segment
If a new event occurs in the curve of one of the above types, the bit corresponding to "Unread_Log_Files_Status_Register" (1,0-0: 96.98.14.255,2) is set to 1,If this curve buffer is read, the corresponding bit is cleared.Remote system can read Unread_Log_Files_Status_Register and determine whether there is a new event occurred

16.3 Event Reporting

Alarm Register 1 (1, 0-0:97.98.0.255);

- Alarm Register 2 (1, 0-0:97.98.1.255);
- Alarm Descriptor 1 (1, 0-0:97.98.20.255);
- Alarm Descriptor 2 (1, 0-0:97.98.21.255);
- Alarm Filter 1 (1, 0-0:97.98.10.255);
- Alarm Filter 2 (1, 0-0:97.98.11.255);

The event reporting uses Data Notification Service of DLMS, when there is an event that needs to be reported, the alarm information is pushed to the system side. Every Alarm Register, Alarm Descriptor, and Alarm Filter are 32bit, and they have exactly the same definition of bits; each one representing an event type;

The above picture is the processing flow of event reporting:
1. Determine whether this type of event needs to be reported according to the configuration in Alarm Filter;
2. When there is an event that needs to be reported, the corresponding bit in Alarm Register is set to be "1".
3. If the corresponding bit of this event is "0" in the previous Alarm Register, then the corresponding bit in the Alarm Descriptor is "1";
4. Send alarm information to the superior system through Data Notification Service

16.3.1 Alarm Register

Alarm Register 1 (1, 0-0:97.98.0.255) of Alarm / Register 2 of (1, 0-0:97.98.1.255) corresponding to 32bitand a total of 64 types of events; Each bit indicates whether the current event is in the state, if the event has ended, then the corresponding bit in the Alarm Register is "0"
In addition, the value in Alarm Register can also be modified through the communication port, but only allowed to write "0".
If the event continues after being cleared, it will be reset to “1”by the meter.

16.3.2 Alarm Descriptor

- Alarm Descriptor 1 (1, 0-0:97.98.20.255);
- Alarm Descriptor 2 (1, 0-0:97.98.21.255);

Alarm Descriptor is used to describe whether the event has been successfully reported, when a bit is set to 1, it will send alarm information to the system side;
After receiving the report, the receiving system needs to write "0" to the corresponding bit in the Alarm Descriptor after receiving the report information, indicating that the report information has been received.

The system can also read the Alarm Descriptor to determine what events are in the reporting state.

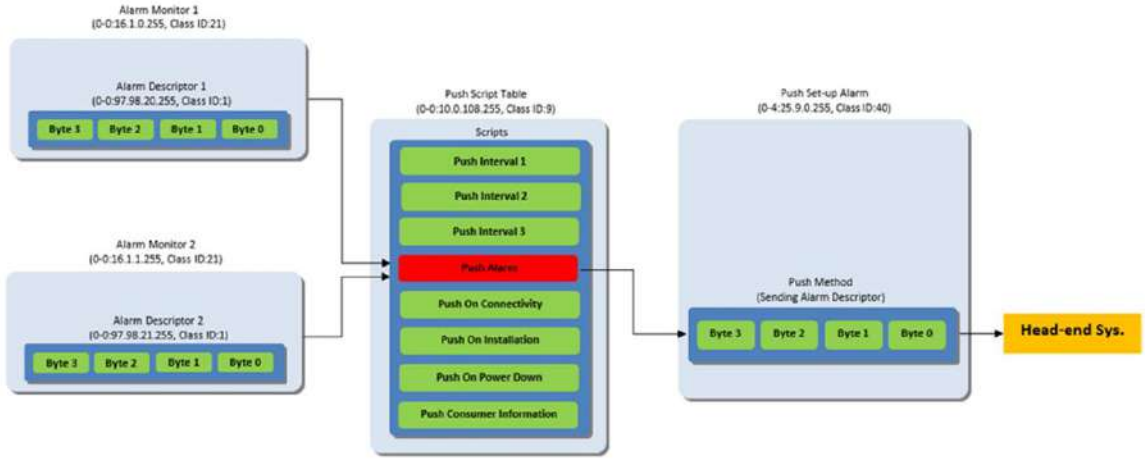
16.3.3 Alarm Filter

- Alarm Filter 1 (1, 0-0:97.98.10.255);
- Alarm Filter 2 (1, 0-0:97.98.11.255);

It is used to configure what events need to be reported.

16.3.4 Event Sending

Event reporting, as shown in Section 3.3, shows that Data Notification Service is only supported in the pre-linked client in default condition;
When events need to be reported, there is no need to establish links.
The following figure shows the event reporting process.



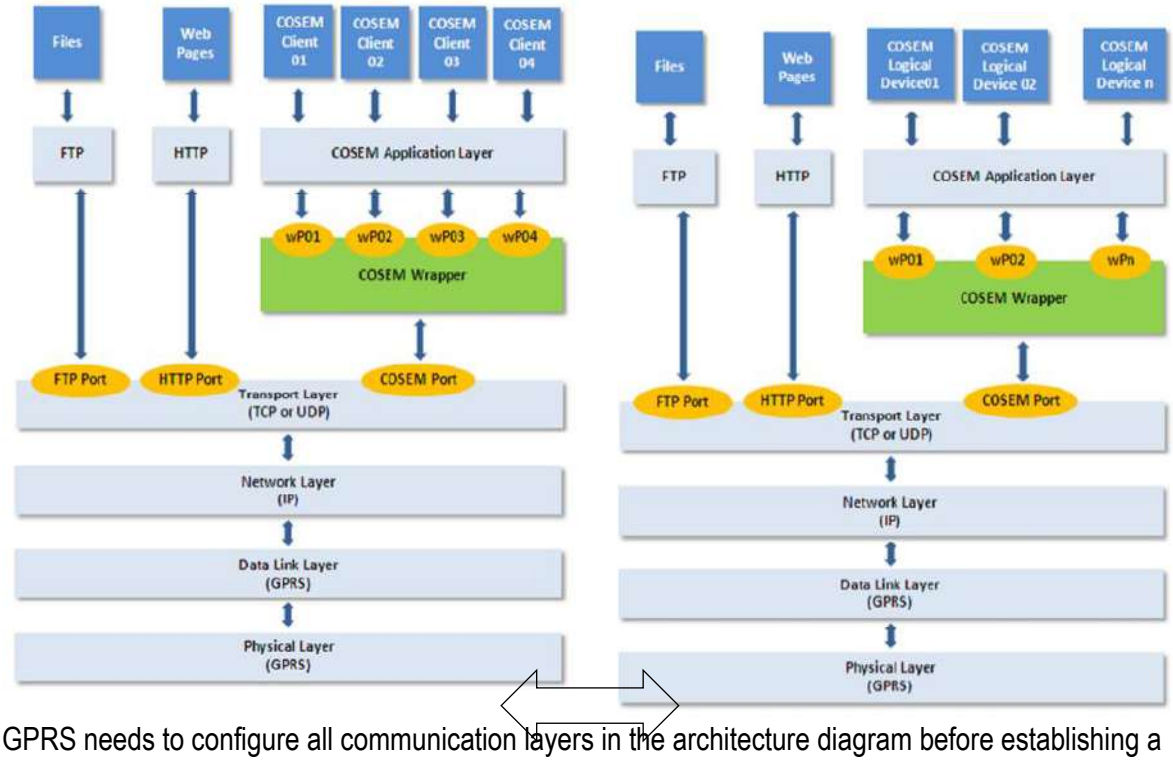
The message of event reporting is sent through the following objects:

- Alarm Monitor 1 (21, 0-0:16.1.0.255);
- Alarm Monitor 2 (21, 0-0:16.1.1.255);
- Push Script Table (9, 0-0:10.0.108.255);
- Push Setup Alarm (40, 0-4:25.9.0.255);

Alarm Monitor 1 (21, 0-0:16.1.0.255,3) and Alarm Monitor 2 (21, 0-0:16.1.1.255,3) monitor Alarm Descriptor 1 (1, 0-0:97.98.20.255,2) and Alarm Descriptor 2 (1, 0-0:97.98.21.255,2) respectively by default.
For Alarm Monitor 1 (21, 0-0:16.1.0.255,2) and Alarm Monitor 2 (21, 0-0:16.1.1.255,2) thedefault monitoring threshold value is "0"; when any value in the Alarm Descriptor is not "0", the default action script Push Script Table (9, 0-0: 10.0.108.255) set in Alarm Monitor 1 (21, 0-0:16.1.0.255,4) and Alarm Monitor 2 (21, 0-0:16.1.1.255,4) is triggerer, message reporting will be activated.
After the system end receives the report message, it will set the alarm bit in Alarm Descriptor to "0", then stop the alarm from generating again;
If for some reason the alarm information is not delivered, the alarm will be resent according to the retry delay Push Setup Alarm(40, 0-4:25.9.0.255,7) and the retry times Push Setup Alarm(40, 0-4:25.9.0.255,6);
The user can set the retry delay and the number of times;

17. GPRS Link

The link of GPRS is handled by the GPRS module, the corresponding class in the electric energy meter can only conduct read and setting, and do not realize the function.
When the GPRS module is powered on, the operation parameters are synchronized to the module. If the parameter changes, it is necessary to call the restart GPRS module script to load the parameters again.Structure is as shown below:



GPRS needs to configure all communication layers in the architecture diagram before establishing a communication link. The objects that need to be configured are as follows:

- Physical Layer Configuration:
  GPRS Modem Setup (45, 0-0:25.4.0.255);

- Data Link Layer Configuration:
  MAC Address Setup (43, 0-0:25.2.0.255);

  PPP setup (44, 0-0:25.3.0.255)

- Network Layer Configuration:
  IPv4 Setup (42, 0-0:25.1.0.255);

  IPv6 Setup (48, 0-0:25.7.0.255)

- Transport Layer Configuration:
  TCP-UDP Setup (41, 0-0:25.0.0.255);

17.1 Physical Layer Configuration(45, 0-0:25.4.0.255)

Configuring the following information by configuring COSEM object GPRS Modem Setup (0-0:25.4.0.255, Class ID 45):
GPRS-APN (Class ID 45, 0-0:25.4.0.255, 2); - If operators do not request, it can not be configured
GPRS-PIN_code (Class ID 45, 0-0:25.4.0.255, 3); - need not to be configured;

17.2 Data Link Layer Configuration

The data link layer contains objects MAC Address Setup and PPP Setup; configuration item:
MAC Address Setup (Class ID43, 0-0:25.2.0.255, 2) - MAC address needs not to be configured
"PPP Setup" -PPP authentication (Class ID 44, 0-0:25.3.0.255, 5) -- only supports PAP authentication, configures user-name and PAP-password, and needs not to be configured according to the operator's requirement.

17.3 Network Layer Configuration

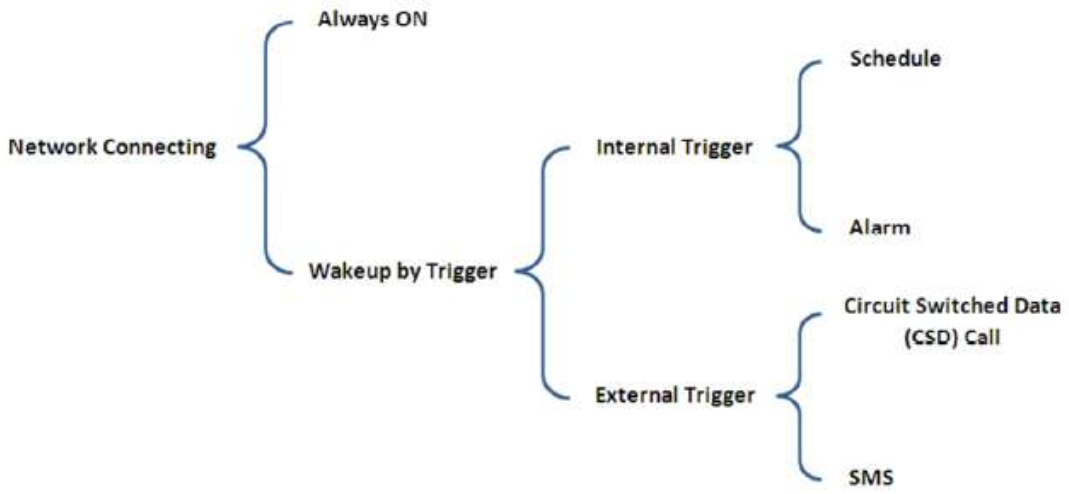
The network layer is configured by class IC 42 IPv4 setup;

Configurable objects:
IPv4 Setup - local IP address (42, 0-0:25.1.0.255, 3); - can be set (static) or automatic acquisition (dynamic);
IPv4 Setup – DNS preferred (42, 0-0:25.1.0.255, 9); - the default is 8.8.8.8, and if a domain name access is required it needs to be set up;
IPv4 Setup - standby DNS (42, 0-0:25.1.0.255, 10); - the default is 0, and if a domain name access is required it needs to be set up;

17.4 Transport Layer Configuration

The transport layer is carried out by TCP-UDP Setup (41, 0-0:25.0.0.255), configurable items:
TCP-UDP Setup - local communication ports (41, 0-0:25.0.0.255, 2) - the default value is 4059, the remote port is consistent with the local;
TCP-UDP Setup – TCP timeout (41, 0-0:25.0.0.255, 6) -- beyond this time if no data is received from the system end then the connection is automatically disconnected. 180 seconds by default, configurable; if configured as 0, then no timeout;

17.5 GPRS Network Connection Mechanism

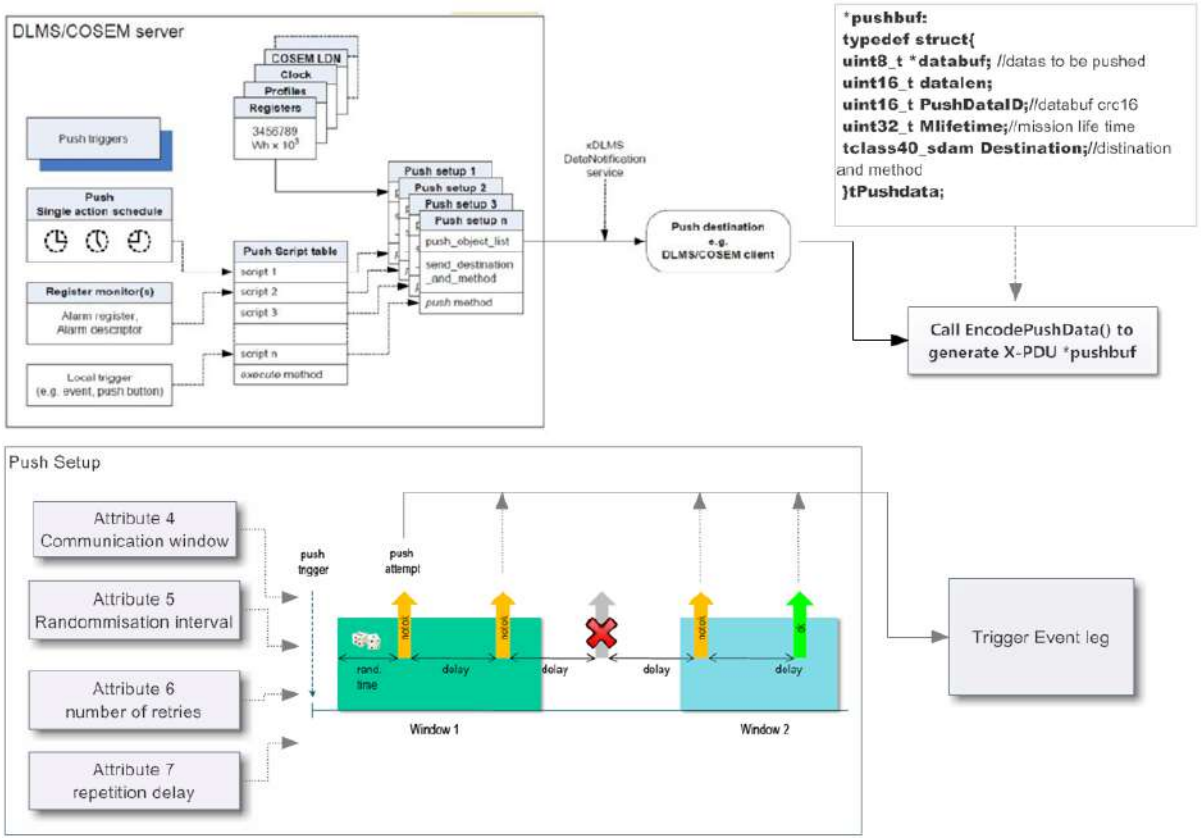


As shown above, GPRS network connection support always online and online wake-up ;Wake-up mode can be internal or external clock, external alarm SMS, call wake (auto answer);
For TCP connections, the device is the client by default and the target address is the server; Network connection mechanism is configured via "AutoreConnect" object (29, 0-0: 2.1.0.255).
"AutoConnect" - mode (29, 0-0:2.1.0.255, 2) - support 0101102103104 five modes can be configured;
"AutoConnect" - reconnect times (29, 0-0:2.1.0.255, 3) - number of reconnect can be configured;
"AutoConnect" - reconnect delay (29, 0-0:2.1.0.255, 4) - reconnect delay time is configurable;
"AutoConnect" - communication time window (29, 0-0:2.1.0.255, 5) - Automatic connection of communication time window, is effective in mode 102103;
"AutoConnect" - connection target (29, 0-0:2.1.0.255, 5) - connected target address, IP address, can be set up to maximum 5 numbers.
Auto Answer - Wake-Up Period (28, 0-0: 2.2.0.255, 3) - Configure the period during which the communication can be woken up;
Auto Answer (CSD Call / SMS) - Number of wake-up rings (28, 0-0: 2.2.0.255, 6) - Number of rings required to configure wakeup;
Auto Answer - Allow Wakeup Number List (28, 0-0: 2.2.0.255, 7) - Configure the communication number that allows wakeup;

The following example illustrates the process of setting the automaticGPRS link connection:
Step1: Set APN: "CNNET"; \
Step2: Set the PPP user name for PAP authentication: "CMC" and password "1234".
Step3: Set TCP local communication port: 4059, TCP timeout: 180S;
Step4: Set the number of AutoConnect reconnection 3, Reconnect delay: 60S;
Step5: Set AutoConnect time window: 23: 00-4: 00 everyday;
Step5: Set the AutoConnect link two target IP addresses: "8.8.8.8", "192.168.1.1";
Step6: Set AutoConnect mode: 103;
Step7: Call the restart GPRS module script (9, 0-0: 10.1.0.255) to resynchronize the parameters;
After setting, the meter will automatically link to the server through the set APN and PPP authentication username and password every day from 23:00 to 04:00.
The preferred destination address is 8.8.8.8. If the link fails, after 60 seconds, try to link 192.168. 1.1, failure repeated up to 3 times; default link port: 4059;
If the system does not receive the data within 180 seconds, the link will be cut off automatically; If the maximum failure in this time window has reached 3 times, then no longer try to link until the next time window again try to take the initiative to link.
At any time, any message, incoming call, scheduled data push or alarm will trigger the automatic link and send the data to the target server. If the connection is not activated within 180s, it will be automatically disconnected if no data is received from the server connection;

18. Active Push (push)

Active push is mainly used for abnormal event reporting and scheduled active data upload. The active push adopts the DLMS / COSEM data notification service to push a set of configured data. The push service processing is performed in an electric meter. If there is a data frame to be sent Communication module directly through the MI1 interface using HDLC UI frame, pushed to the module, the module cache push data and complete the sending; the module can cache the maximum 3 frames push data, please refer to the specific package 2; Push service processing flow is as follows Figure:



As described in Chapter 18, if the device can wake up the remote data link function (such as GPRS) through event reporting or scheduled data push to send data to the remote server, the wake-up function can be configured through the following objects:

- Push action scheduler – Interval_1(0-1:15.0.4.255, Class ID: 22);
- Push action scheduler – Interval_2 (0-2:15.0.4.255, Class ID: 22);
- Push action scheduler – Interval_3 (0-3:15.0.4.255, Class ID: 22);
- Push Setup- Interval_1 (0-1:25.9.0.255, Class ID: 40);
- Push Setup- Interval_2 (0-2:25.9.0.255, Class ID: 40);
- Push Setup- Interval_3 (0-3:25.9.0.255, Class ID: 40);
- Alarm Monitor 1 (0-0:16.1.0.255, Class ID: 21);
- Alarm Monitor 2 (0-0:16.1.1.255, Class ID:21);
- PushSetup-On Connectivity (0-0:25.9.0.255, Class ID: 40);
- PushSetup-On-Alarm (0-4:25.9.0.255, Class ID: 40);
- PushSetup- On-Installation (0-7:25.9.0.255, Class ID: 40);

18.1 Scheduled Trigger Push:

According to the project requirements, you can configure different scheduled push frequency and push content, the following objects can be used to set scheduled push:

- Push action scheduler-Interval_n execution time (22,0-n: 15.0.4.255, 4); - Configure push time
- Push Setup- Interval_n Push object list (40, 0-n: 25.9.0.255,2); - Configure push content, default null
- Push Setup- Interval_n push target IP address (40, 0-n: 25.9.0.255,3); - configure the target IP address
- Push Setup- Interval_n Push Random Window Time (40, 0-n: 25.9.0.255,5); - Configure Random Window Time
- Push Setup- Interval_n Number of push retries (40, 0-n: 25.9.0.255,6); - Number of retries configured
- Push Setup - Interval_n Push Retry Delay (40, 0-n: 25.9.0.255,7); - Configure Retry Delay

For the scheduled trigger push, the system does not reply; the meter is pushed only once and is processed by the module successfully; the methodsof different communication modules are greatly different in the way of ensuring the success of the push;

18.2 Alarm Trigger Push

Section 17.3 shows how to trigger the message push mechanism after an event occurs.
For the push triggered by the alarm, after the system receives the alarm event, the bit corresponding to the Alarm Descriptor in the device is set to 0.
The device determines whether the alarm has been successfully uploaded by judging the Alarm Descriptor, so as to trigger the resend times and resend Delay mechanism

18.3 GPRS Connection Trigger Push

When the device detects that a new TCP connection is set up (GPRS), it triggers the automatic registration information push.
This connection can be done through the scheduled automatic connection as described in chapter 18.5, or it can be a wake-up link. This part of the function is conducted by the energy meter.
After detecting a new GPRS connection, the GPRS module sends an action command (9, 0-0.10.0.108.255) to the watthour meter through the MM1 interface, script_identifier = 5, and triggers the push setup on connectivity service.
The wake-up registration configuration can be done with the following objects:

- Push Setup-On Connectivity - push object list (40, 0-0: 25.9.0.255, 2) -; to configure pushed objects;
- Push Setup-On Connectivity - push destination IP address (40, 0-0: 25.9.0.255,3); - configure destination IP address
- Push Setup-On Connectivity - push random window time (40, 0-0: 25.9.0.255,5); - configure

random window time

- Push Setup-On Connectivity - push retries (40, 0-0: 25.9.0.255,6); - configure retries
- Push Setup-On Connectivity - push retry delay (40, 0-0: 25.9.0.255,7); - configure retry delay

For Push Setup-On Connectivity, the system end does not reply; if the device receives the push of the DLMS data from the system side to trigger resend times and resend delay mechanism;
By default, the system reads the Logical Device Name (1, 0-0: 42.0.0.255, 2) of the energy meter after the push message is received to maintain the link.
Default push content:

- Logical Device Name (1, 0-0:42.0.0.255, 2);
- IPv4 setup-GPRS (42,0-0:25.1.0.255, 3);

19. Security Mechanism

The device supports low level and high level two types of security mechanisms. The default factory configuration is low level. When the security level needs to be increased, the device does not need to be securely encrypted. To reduce the security level, you must encrypt the data Can be set by the following objects:

- Security Setup (Management Client/Pre-Establishment Client) (64, 0-0:43.0.0.255,2)
- Security Setup (Reading Client) (64, 0-0:43.0.2.255,2)
- Security Setup (Consumer Information) (64, 0-0:43.0.1.255,2)-CIP

The default factory key is as follows:

Security parameter	Default value (hex)
LLS default password	12345678
Global Authentication key	0xD0 D1 D2 D3 D4 D5 D6 D7 D8 D9 DA DB DC DD DE DF
Global Broadcast key	0x0F 0E 0D 0C 0B 0A 09 08 07 06 05 04 03 02 01 00
Global Encryption key	0x00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
Global CIP Authentication key	0xC0 C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF
Global CIP Encryption key	0x10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
Global Reading Authentication key	0 x E0 E1 E2 E3 E4 E5 E6 E7 E8 E9 EA EB EC ED EE EF
Global Reading Encryption key	0 x F0 F1 F2 F3 F4 F5 F6 F7 F8 F9 FA FB FC FD FE FF
Master key	0X00 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF

Security Setups	Keys
Security Setup (Management Client/Pre-Establishment Client) (64, 0-0:43.0.0.255)	Unicast Global Key Broadcast Global Key Authentication Key Master Key
Security Setup (Reading Client) (64, 0-0:43.0.2.255)	Unicast Global Reading Key Authentication Reading Key
Security Setup (Consumer Information) (64, 0-0:43.0.1.255)	CIP Unicast Global Key CIP Authentication Key

For communication on the local port, you can limit the number of times the wrong authentication allowed and the time that the port is locked after the maximum number of times is reached by setting the Local Authenction on protection object (1,0-0: 96.98.20.255.2)
structure
{
unsigned: failed attempts (1-10 times)
unsigned: lockout time (1-180 minutes)

}

When communicating using the Broadcast key or unicast key, the device records frame counters in the following objects:

- Security - Receive Frame Counter - Unicast Key (Remote) (0-0:43.1.0.255);
- Security - Receive Frame Counter – Broadcast Key (Remote) (0-1:43.1.0.255);
- Security - Receive Frame Counter - Unicast Key (Local) (0-0:43.0.2.255)；

Annexure- C

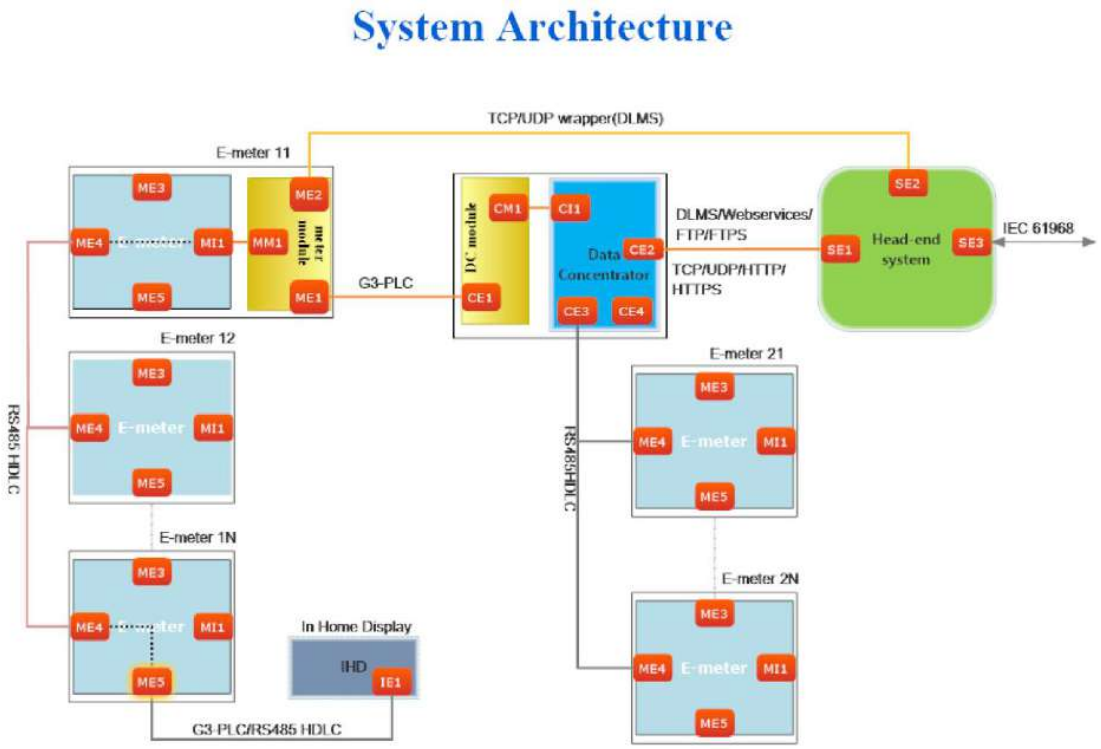
Unified Meter Data Collection System

DCUto HES

1. General

1.1 Introduction

The system interface architecture of the device is shown in the following figure; the electric energy meter, concentrator and front end system are simplified into several interfaces marked with correspondent code name respectively;



Device functions & Interoperability Interfacing Requirements are composed of 3 parts, including:

Package 0： Addressing System

Package 1： DCU to HES

Package 2： Meter Modeling

1.2 Reference Documents

Name
Blue_Book_12th_edition
Green_Book_8th_edition
Object_defs_v3.1_161215

2. Communication Structure

Above figure is direct communication structure between HES and DCU, which contains 3 independent channels. In specific, the system itself is featured by its high flexibility and

compatibility for various requirements. Interface CE2-1,CE2-2， CE2-3 of DCU could be configured to remote port with network at will.

As shown in the structure diagram, data exchange between data collection system and devices to be read supports 3 standard communication models:

- DLMS in IEC 62056
- WebService
- FTPS/FTP

3. DLMS

3.1 DLMS Communication Model

The data exchange between the data acquisition system and the collected equipment, should refer to the communication mode of the client / server AP specified by DLMS. The client AP initiates the access request and the server AP responds, which is a question and answer communication mechanism; and the role of server AP is specified to be assumed by the meter or module.

By means of access management, one server AP can exchange data with one or more client APs at the same time, and a client AP can also access one or more server APs at the same time.

In addition to the normal question and answer mechanism, if there are abnormal events in the server AP, the event can be reported through the prescribed format.

The three-layer architecture of DLMS is shown in the following figure:

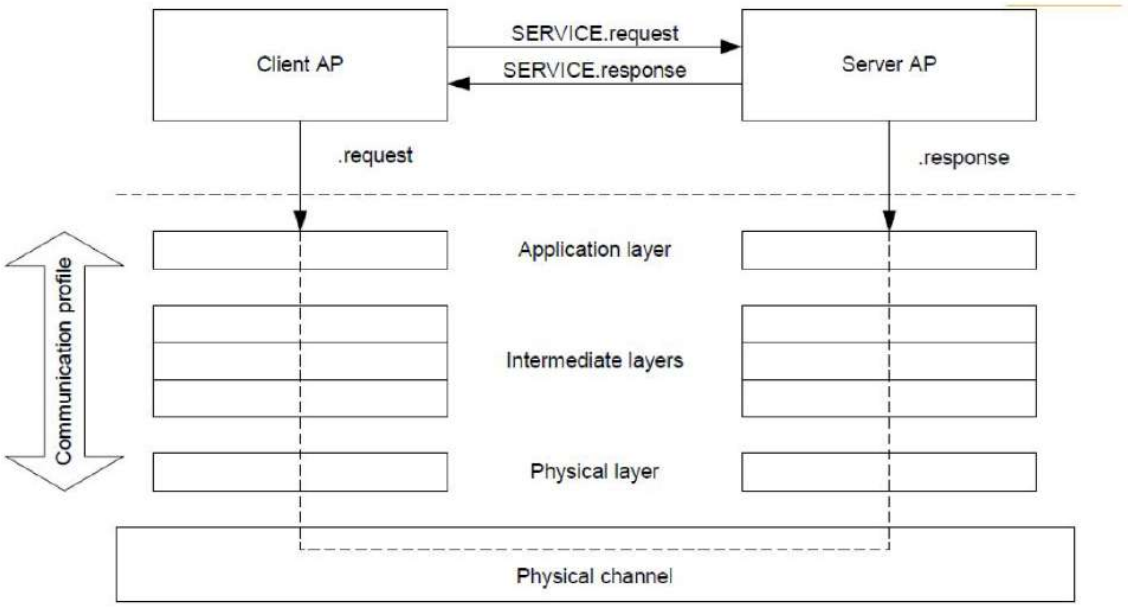


Figure 2 – Client/server relationship and protocols

Level 1: Physical Device Layer

Level 2: Logical Device/Intermediate Layer

Level 3: COSEM Application Layer

3.2 DLMS Application Layer Service

Application layer service default settings is suporting LN(logical Name) only;

ContextName = LONG_NAMES;

Version No.:

DLMS version = 6;

According to DLMS standard, meter supports service types as below chart:

Service	Conformance Block Bit
general-protection	1
general-block-transfer	2
block-transfer-with-get	11
block-transfer-with-set	12
multiple-references	14
data-notification	16
Get	19
Set	20
selective-access	21
action	23

For different clients, defaulted services supported are as below chart:

Client	Function
Public Client	Get block-transfer-with-get
Pre-linked Client	get set action data-notification general-block-transfer general-protection
Management Client	get block-transfer-with-get set block-transfer-with-set selective-access multiple-references action general-block-transfer general-protection
Read Client	get block-transfer-with-get selective-access multiple-references general-block-transfer general-protection
Module Client	get block-transfer-with-get set block-transfer-with-set selective-access multiple-references

	action general-block-transfer general-protection

For different client No. of each logical device, there is an application layer connection subject.Class 15 （Association LN） is used for describing present connection information including connection list and connection status, etc. Moreover, it could also modify key through Class 15 with HLS verification, connection subject list as as below:
“Current Associaon” (0-0:40.0.0.255, Class ID: 15)-present connection subject
“Public Associaon” (0-0:40.0.1.255, Class ID: 15)
“Management Associaon” (0-0:40.0.2.255, Class ID: 15)
“Pre-established Associaon” (0-0:40.0.3.255, Class ID: 15)
“Reading Associaon” (0-0:40.0.4.255, Class ID: 15)
“Module Associaon” (0-0:40.0.5.255, Class ID: 15)

3.3 Wrapper

3.3.1 Wrapper Format

Wrapper includes 8 bytes for version, SSAP, DSAP and LEN. Frame format as below:

Version	SSAP	DSAP	LEN	COSEM APDU

■Parameters Descriptions:
Version: 2bytes, Wrapper version, present version is 0x0001;
SSAP: 2 bytes, refer to sender SAP address;
DSAP: 2 bytes, range 0x0001 – 0xFFFF, refer to destination address;
LEN: 2 bytes, range 0x0000 – 0xFFFF, refer to bit length of COSEM APDU transmitted;
COSEM APDU: communication data content, out of range of wrapper.

3.3.2 SSAP Client Address

SSAP length is 2 bytes, refer to client address, defined as below:

High Byte	Low Byte

High byte: master station address range 0x01 – 0xFF refer to various transformer master station address, 0x00 means not specified master station address (default);
Low byte: client address 0x10/0x02/0x01/…, refer to public/read/write/..client id, SSAP

3.3.3 DSAP Service Address

DSAP length is 2 bytes, refer to service address; mainly refer to device address accessed, division of DSAP is mainly used to identify logical device of the accessed device.

3.4Gateway Protocol

Gateway protocol is mainly use for gateway devices (DCU/collector/communication module), which supports multiple layer network transparent transfer, as shown in below figure:

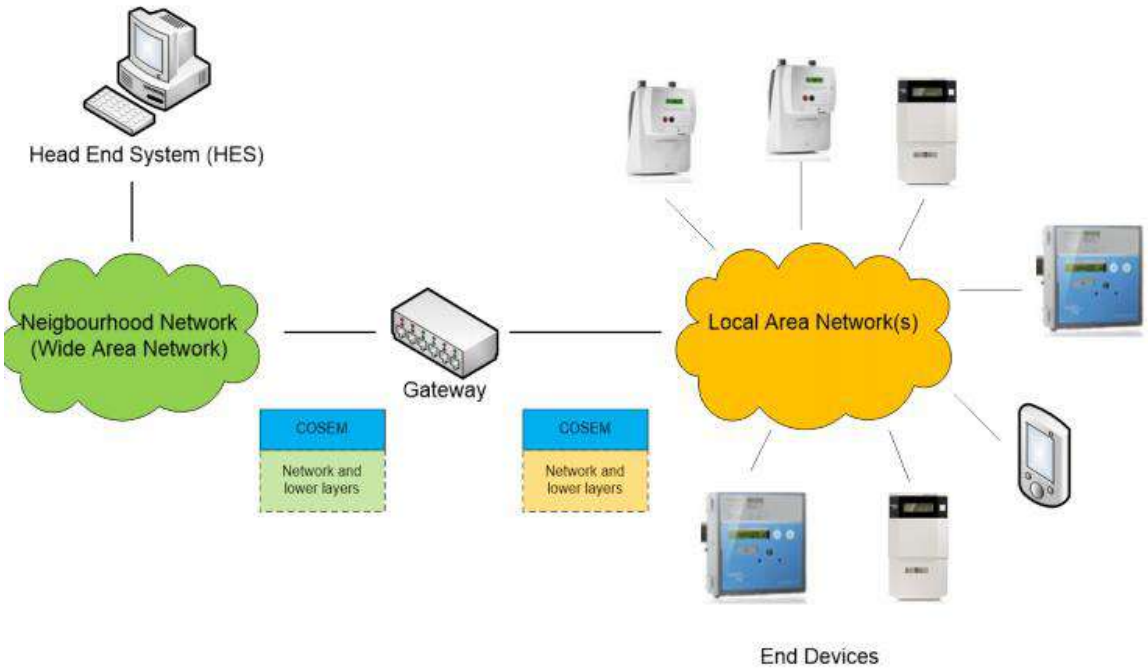


Figure 156 – General architecture with gateway

3.4.1 Gateway Protocol Format

Gateway protocol definition as below, Prefix for short;

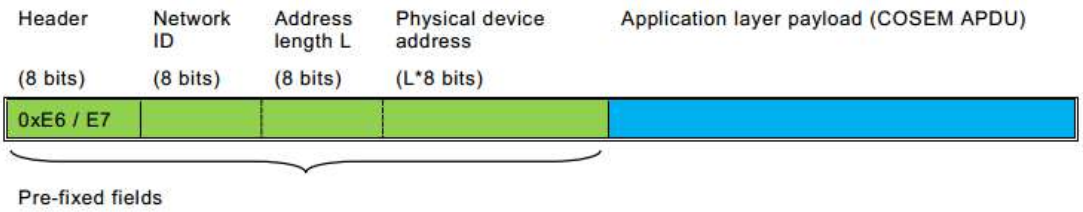


Figure 157 – The fields used for pre-fixing the COSEM APDUs

■Parameter Description:

Header: 1 byte, indicating the data transfer direction, 0xE6 is a request or data report frame , 0xE7 is a response frame;

Network ID: 1 byte, network ID, indicating the forwarding of the lower channel number (network number);

Address length: 1 byte, target address length;

Physical device address: N bytes, the physical address of the target,
whose length depends on the address length;

COSEM APDU： The communication data content, does not belong to the gateway protocol.

3.4.1 Gateway Protocol Address

3.4.1.1 Network ID

Network ID		
NO	Address	
1	0x0000	Default fixed port (1 port only)
8	0x0041	XE1 port (module 1, PLC /RF/…);
9	0x0042	XE2 port (remote ports: 4G/3G/GPRS/Ethernet);
10	0x0043	XE3 port (RS485 1);
11	0x0044	XE4 port (RS485 2);

12	0x0045~0x006F	Port reserved

3.4.1.2 Physical device address

Gateway protocol destination address definitions as below:

Address length: 0x07； 7bytes physical address;

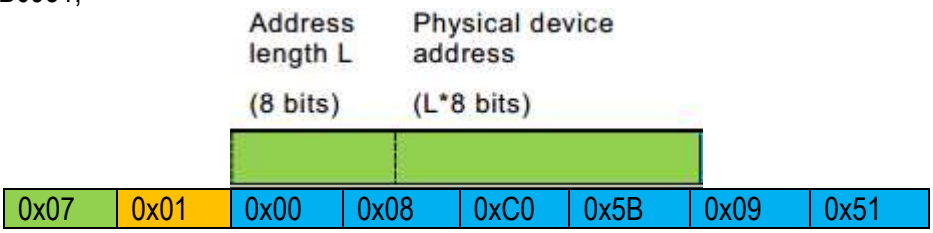
Physical device address: 7bytes.

1 byte, logic address;

Physical device upper address		
NO	Address	
1	0x00	No-station
2	0x01	Main logical device/(DCU/meter/collector)
3	0x02~0x0F	DLMS UA reserved
4	0x10	Communication module 1 function area
5	0x11	Communication module 1 firmware area
6	0x10~0x001F	Main logical device reserved
7	0x22~0x3F	DC reserved
8	0x41	XE1 port (module 1, PLC /RF/…);
9	0x42	XE2 port (remote ports: 4G/3G/GPRS/Ethernet);
10	0x43	XE3 port (RS4851);
11	0x44	XE4 port (RS4852);
12	0x45~0x6F	Port reserved

6 bytes, target physical address; product serial No. in 16 carry scale code;

E.g.: Device serial No.: 037586930001, then physical address in gateway protocol shall be 0x0008C05B0951;



3.5 DLMS Device Identification

Belows are for description about how devices are defined according to DLMS/COSEM requirements. Each device has an unique ID (serial No.) correspondently, which provides proof for registration of device in HES system;

Device ID： “Device ID7” COSEM object (1-0:0.0.0.255， Class ID: 1)； this ID is unique, default in 11 digits, coding may be changed as per utility’s requirement, supports 14 digits in maximum. The ID is set in device at factory settings by manufacturer and shall be labled on device nameplate;

System Title： As per requiremetn of various projects, each device has an unique 8 bytes system title/code, which has been writen in device firmware at factory settings; it facilitates identification and verification of various manufacturers or device models in the same system.

Logical Device Name: “COSEM Logical Device Name” COSEM object (0-0:42.0.0.255， Class ID: 1). As mentioned in above context, each physical device may correspond to multiple logical devices, logical devices name is the unique ID for identify logical device.

3.6 DLMS Device Automatic Registration

Device automatic registration refers to the process of sending device logical device name, IP address to HES through Data-notification service after device installed and HES responses with registration completion after receiving.

DLMS device automatic registration refers to configuration contents of subjects which adopted push setup –on installation (0-7.25.9.0.255, class 40) send registration information to the upper layer devices by Data-notification service. Default settings in device supports pre-linked Client for Data-notification service only.

Upper layer devices shall response after receiving registration information and confirm, and then set "E_instal" information to data item (0-0:96.13.1.255, class 1).

Press device scroll display button for 10 seconds, (LCD shows "install_S"), then trigger device automatic registration report; it also support configuration by push setup –on installation for automatic registration time and tryout times, etc.

In default settings, subjects are automatically registered and sent:

Logical Device Name: (1, 0-0:42.0.0.255，2);

IPv4 Setup –local IP address (42, 0-0:25.1.0.255, 3);

4. Webservices

Webservice is realized in SOAP 1.1 format. SOAP Information transmission process from HTTP adopts HTTP over TLS1.2(HTTPS) for data encryption;

4.1 Webservices

HES could make main operations through webservices provided by DCU. All configuration and reading commands have user name and password protection. User name and password are configurable through local programming or through higher authorization in DLMS. Commands are as below:

Default User Name：DefaultUser

Password：helloworld

4.1.1 GWTransReq

"**GWTransReq**"：Gateway transmission request is used for transparent data transmission to down layer devices under each ports, response with "**GWTransResp**" for settings; After DCU completes execution of command, then inform HES with "**GWTransInfo**".

4.1.2 ConfigDCUcom

"**ConfigDCUcom**"： set DCU for modification of meter reading configurations, such asDLMS Client No., HES address, etc. After configuration, response with "**ConfigDCUcomResp**" to HES.

4.1.3 GetConfigDCUcom

"**GetConfigDCUcom**": Acquire DCU current operation parameters, shall correspondent with contents in "ConfigDCUcom", then DCU response "**GetConfigDCUcomResp**" with correspondent parameters；

4.1.4SetDatatime

"**SetDatetime**": set DCU current time; If NTP server is set, DCU would automatically synchronize clock, but HES could also directly set DCU clock through this service, then response with "**SetDatetimeResp**"

4.1.5GetDatatime

"**GetDatetime**": Acquire DCU current time; If NTP server is set, DCU would automatically synchronize clock, but HES could also directly set DCU clock through this service, then response with "**GetDatetimeResp**".

4.1.6SetDeviceDoc

"**SetDeviceDoc**": Acquire or modify meter data under all or certain port, then response with："**SetDeviceDocResp**".

4.1.7GetDeviceDoc

"**GetDeviceDoc**": Acquire all meter data information and its correspondent address under each port, then response with "**GetDeviceDocResp**".

4.1.8GetOnlinelist

"**GetOnlinelist**"：Acquire currently online meter list of certain port, then response with "**GetOnlinelistResp**".

4.1.9AMRTask

"**AMRTask**": Set automatic meter reading task of each port, able to set multiple automatic meter reading task, daily/monthly/customized cycle and select meter reading item list to place commands; then response with "**AMRTaskResp**".

4.1.10GetAMRTask

"**GetAMRTask**": Acquire automatic meter reading task of each port and response with "**GetAMRTaskResp**".

4.1.11RestartAMRTask

"**RestartAMRTask**": Freeze single or multiple automatic meter reading task and restart, then response with "**RestartAMRTaskResp**".

4.1.12GetDeviceData

"**GetDeviceData**": Acquired device data from DCU after executing "AMRTask"; able to acquired single or all data, then response with "**GetDeviceData**".

4.1.13DeviceUpgradeTask

"**DeviceUpgradeTask**": Set upgrade tasks for devices managed by DCU. After configuration, DCU shall download upgrade file in specific time from FTP and execute the task, then response with "**DeviceUpgradeTaskResp**"； After execution, DCU shall inform HES through "**DeviceUpgradeTaskInfo**".

4.1.14GetDeviceUpgradeSta

"**GetDeviceUpgradeSta**": Acquire meter upgrade task execution status and response with"**GetDeviceUpgradeStaResp**".

4.1.15DCUUpgradeTask

"**DCUUpgradeTask**":Set upgrade tasks forDCU. After configuration, DCU shall download upgrade file in specific time from FTP and execute the task, then response with"**DCUUpgradeTaskRes**"; After upgrade, DCU shall inform HES "**DCUUpgradeTaskInfo**".

4.1.16SchduleTack

"**ScheduleTask**": Configure schedule task in single or multiple commands are available;
"**ScheduleTaskInfo**": After task completed, DCU shall inform HES "**ScheduleTaskResp**".

4.2 XML Example

GWTransReq: TaskNum:00-99 Protocoltype:0:DLMS

```
<?xml version="1.0"?>
<soapenv:Envelopexmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/ "
xmlns:lh="http://locahost:8086/DcuSoap">
<soapenv:Body >
<lh:GWTransReq UserID=”DefaultUser” Password=”hellowrold”  PortID=”CE1”
DeviceID=”37999900001”>
<TaskNum>01</ TaskNum >
<!--one or more repetitions:-->
<Trans>
<Protocoltype>0</ Protocoltype >
<Databody>(encrypted COSEM)</ Databody >
</Trans>
<Trans>
<Protocoltype>0</ Protocoltype >
<Databody>(encrypted COSEM)</ Databody >
</Trans>
   </lh: GWTransReq >
</soapenv:Body>
```

```
</soapenv:Envelope>
```

DCU shall inform HES through GWtransInfo after completing task;
GWTransInfo: TaskNum:00-99 TransResult:0:succeed others: Error Code

```
<?xml version="1.0"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/ "
xmlns:lh="http://locahost:8086/DcuSoap ">
<soapenv:Body >
<lh:GWTransInfo UserID=”DefaultUser” PortID=”CE1” DeviceID=”37999900001”>
<TaskNum>01</ TaskNum >
<!--one or more repetitions:-->
<TransResp>
<TransResult>0</ TransResult >
<DataReap>(encrypted COSEM)</ DataReap >
</ TransResp >
< TransResp >
<TransResult>0</ TransResult >
<DataReap>(encrypted COSEM)</ DataReap >
</ TransResp >
   </lh: GWTransInfo>
</soapenv:Body>
</soapenv:Envelope>
```

SetDatatime :TaskNum:00-99

```
<?xml version="1.0"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/ "
xmlns:lh="http://locahost:8086/DcuSoap ">
<soapenv:Body >
<lh:SetDatatime UserID=”DefaultUser” Password=”hellowrold” >
<TaskNum>01</ TaskNum >
<Datatime>2018-01-02F00:00:00Z</ Datatime >
   </lh: SetDatatime >
</soapenv:Body>
</soapenv:Envelope>
```

DCU shall inform HES in SetDatetimeResp after execution;
SetDatatimeResp:TaskNum:00-99 Result:0:succeed others:error code

```
<?xml version="1.0"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/ "
xmlns:lh="http://locahost:8086/DcuSoap ">
<soapenv:Body >
<lh:SetDatatimeResp UserID=”DefaultUser”>
<TaskNum>01</ TaskNum >
<Result>0</Result >
   </lh: SetDatatimeResp>
```

```
</soapenv:Body>
</soapenv:Envelope>
```

For more examples, please refer to WSDL document.

5. FTP/FTPS

FTP/FTPS service is applicable for DCU remote upgrade. HES may configure FTP server address through webservice or DLMS protocol. DCU shall download update files from FTP service in specified time to finish upgrade during execution of upgrade task.

FTP/FTPS service is also applicable schedule automatic meter reading data upload to server. After finishing task, configuration completed.

Annexure- D

Unified System &HES Interface Specification

CIM based InteroperableStandard

1. Overview

This document specifies an implementation profile for Unified System and MDC using common information model (CIM). CIM(IEC61968) mandates how the various messages may be transported from a system to another system.The current version of CIM deals with webservices and JMS as the barriers of transported messages, This document describes how the message payloads are conveyed using web services. The goal is to provide the sufficient details which are needed for the interoperable implementation of CIM. To this aim, the communication architecturre is proposed and the required communication interfaces are specified.

2. Communication Architecture

2.1 Introduction

MDC and an Unified System provide two service interfaces, depicted in Figure 1.1. Each MDC provides an interface called “doCommand”, to serve the operation and data collection commands from its corresponding Unified System. The synchronous responses to the commands are sent to the Unified System as a return value of calling doCommand. An Unified System may ask for an asynchronous response to its sent command by setting the tag of “AsyncReplyFlag" True in header of its input command. In this case, the asynchronous response is sent to the Unified System via “getResponse” service.

Note that, in case of asking for an asynchronous response, the return value of “doCommand” is a xml formatted data which contains the acknowledgement response and the business-level response which will be sent later to the Unified System by calling “getResponse” service.

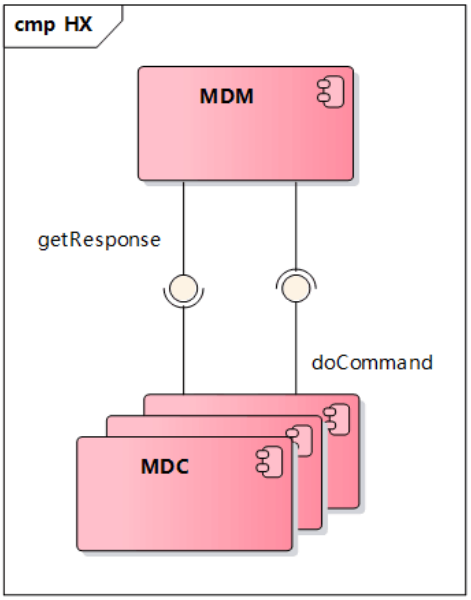


Figure1.1Communication Architecture

2.2 Service Interface

As mentioned, two different interfaces are provided by MDC and Unified System, described in the following table. In the both of the interfaces the types of the input parameters and the output results are native “string”. The contents of the input and the output strings conform to IEC61968 standard.

Table1.1: Service Interface Specification

Service Name	Provider	Input	Output	Description
doCommand	MDC	A string parameter in XML format, conforming to IEC61968 standard for requests.	A string output result in XML format, conforming to IEC61968 standard for responses.	The commands are sent to MDC via this interface and the synchronous responses are returned as the service calls’ results.
getResponse	Unified System	A string parameter in XML format, conforming to IEC61968 standard for responses.	No output(void type)	This command receives theasynchronous responses which are sent by MDC.Unified System

doesn't need to respons
any message to MDC.

3. Transporting Message

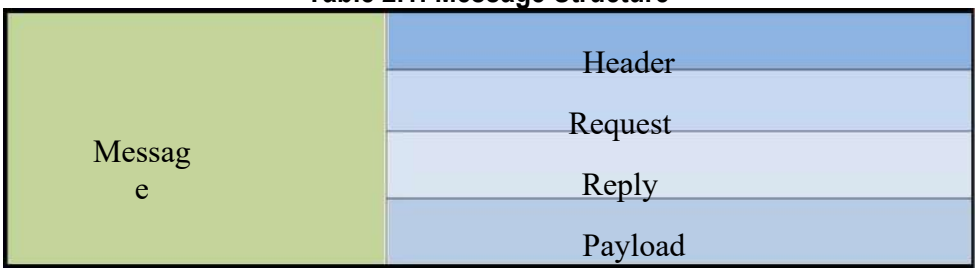
3.1 Introduction

In accordance to this document, in order to establish interoperability between Unified System and MDC of meter manufactures, messages structure should be complied with IEC61968 standards. Since this standard is an open standard, there should be some kind of consensus between manufactures.After reaching consensus regarding message structures, MDC should be tested to verify the accuracy of the implementation.

Manufactures should be considered following points:

Messages should be completely based on IEC61968 and there should be no innovation in messages.
All elements used in the messages should be based on the docment.
According to IEC61968, general structure of messages is defined in the following tables:

Table 2.1: Message Structure



In the above structure, the presence of "Header" is Mandatory and its format is defined inTable 2.2:

Table 2.2: Format of the Header

Header		
Elements	Content	
Verb	Identifies the action of the source Valid value: refer to other Section	
Noun	Identifies the message type Valid value: Refer to other Section	
Revision	Identifies the XSD version for the inbound meter reads Valid value: 2.0	
Context		
Timestamp	Time of sending the message by sender	
Source	Source of Data Unified System/MDC(sender)	
AsyncReplyFlag1	true/false2	
ReplyAddress	Call back URL. String	
AckRequired3	true/false2	
User1	UserID	string
	Organization	
MessageID	UUID01	
CorrelationID	Request Message ID	
Comment	String	

Header		
Elements		Content
Property4	Name	"timeout(s)"/"timeout(m)"/"timeout(h)"
	Value	integer
Property5	Name	"GetConnectedToTheMeter"
	Value	true/false*****
Property6	Name	"password"
	Value	string
1-This element isused just in request messages which are sent by Unified System. 2-It is true/false if the transaction is asynchronous/synchronous. 3-It is only used when the transaction is asynchronous, true means MDC need to return Acknowledgement to Unified System, false means no need. 4-This node can be used for identifying the amount of timeout for request messages. 5-This element is used when collecting data from the meter is required, not from server's database. (This element is not usually used.) 6-This node along with "User" node can be used Authentication process for the messages.		

According to the IEC61968, two kinds of message is defined: request message, responsemessage.Any IEC 61968-9 message, whether it is a request message, response message, is composed as an XML document. These different message types aredistinguished by the top-level element in the XML document. This must always be one of<RequestMessage>, <ResponseMessage>accordingly.

3.2 Request Message

Request messages are used for sending queries or commands. For example, a request message might be sent from a Unified System to an MDC to obtain a set of meter readings.Response messages are used for returning the corresponding data or status information to a request message.
The elements used in the request message are observed in Table 2.4:

Table 2.4: Request Message

Request Message	Header	According to Table 2.2
	Payload	According to verb and noun

3.3 Response Message

Response messages are also used to indicate whether a given request succeeded or whether there were any failures in performing the command. In an asynchronous way, response messages will be used for sending simple acknowledgements in the context of webservices, MDC sends an acknowledgement to Unified System indicating that it received the message, MDC then goes away and in its own time performs a session with the meters of interest, reads the values and when it is ready, sends it back to Unified System in response message.
As mentioned before, in an asynchronous message, there is a "response message" besides the acknowledgement. More details regarding the "Simple Acknowledgement" message is provided in Table 2.5:

Table 2.5: Simple Acknowledgement Message

Response Message (Acknowledgement)	**Header**	**According to Table 2.2**	
	Reply	Result	OK
		Error code	0.3

The response message is defined with more details in Table 2.6:

Table 2.6: Response Message

Response Message	**Header**	**According to Table 2.2**
	Reply*	Result OK/Failed/Partial** Error code RefertoTable2.8
	Payload***	More details in next sections

*This node is used in every response message.
**If error code is 0.0 the result isOK. Other Error codes are provided inTable2.7.
***Payload is only used when data is transferred.

Details for Reply element of response message are provided in Table below:
Note that there may be multiple Error node, showing different error.codes.

Table2.7: Reply

Reply	Result	OK/PARTIAL/FAILED	
	Error (Unbounded)	Code	RefertoTable2.8
		Level	INFORM/WARNING/FATAL/CATASTROPHIC
		reason	
		details*	String
	operationId**		

*In this node, more detail about the error is provided.
**This is used for identifying the number of data packet in a PARTIAL reply, i.e. the Result=PARTIAL and operationId identifies which number of data packet is being sent.

In the following, there are some error codes which are sent from MDC in response to Unified System request:

Table 2.8: Error Codes

No.	Code	Description
1	0.0	Successful/no error
2	0.1	Partial result(additional results conveyed in separate messages)
3	0.2	Partial result(no further results to follow)
4	0.3	Acknowledgement
5	2.4	Invalid meter(not registered)
6	2.6	Invalid ReadingType
7	2.5	Invalid Noun
8	2.9	Invalid Verb
9	2.12	Invalid Usagepoint(s)
10	2.13	Meter/Usage Point mismatch
11	2.14	Invalid source
12	2.15	Invalid RequestID
13	2.33	Invalid PricingStructure(s)
14	3.2	Too many pending requests
15	4.1	Request timedout
16	4.2	Service not available
17	4.3	Local error in processing
18	5.3	Unable to process the request-transaction attempted and failed
19	5.5	Some or allof the requested readingtypes are unavailable in Unified SystemS
20	5.7	Some or all of the requested data is unavailable
21	5.8	Unable to process the request–.Mandatory

		field(s) missing
22	6.5	Request canceled by user
23	7.4	Authentication failure
24	7.5	Action not authorized for user
For other Reply Codes refer to annex B of IEC61968-9		

4. Meter Archive Synchronization

4.1 Introduction

This section aims at presenting message structures regarding adding an equipment.

4.2 Usecase Description

Firstly Unified System should be to validating the meter's information. After thatUnified System will send the command of synchronizing the meter to MDC which contains meter's full profile. The details of Unified System's request to MDC are available in the following Tables:

Table 4.1: Meter Archive Synchornization (add meter)

Meter Addition			
Message type	Verb	Noun	Main elements
Request	create	MeterConfig	Header/Payload
Response	reply	MeterConfig	Header/Reply

Note:
This transaction can be conducted only in synchronous modes. In synchronous mode, a reply will sent back to Unified System, showing the successfulness/unsuccessfulness.

Table 4.2: Meter Archive Synchornization (modify meter)

Meter Modification			
Message type	Verb	Noun	Main elements
Request	change	MeterConfig	Header/Payload
Response	reply	MeterConfig	Header/Reply

Note:
This transaction can be conducted only in synchronous modes. In synchronous mode, a reply will sent back to Unified System, showing the successfulness/unsuccessfulness.

Table 4.3: Meter Archive Synchornization (delete meter)

Meter Deletion			
Message type	Verb	Noun	Main elements
Request	delete	MeterConfig	Header/Payload
Response	reply	MeterConfig	Header/Reply

Note:
This transaction can be conducted only in synchronous modes. In synchronous mode, a reply will sent back to Unified System, showing the successfulness/unsuccessfulness.

Payload's elements regarding this request are presented in Table 4.4:

Table 4.4: MeterConfig

Payload of MeterConfig				
MeterConfig	Meter (unbounded)	mRID		Meter ID
		Names	name	**Serial Number**
			NameType	name **"serialNumber"**
		Names	name	**Meter Model (HX110-KP,INHE110-SP)**
			NameType	name **"meterModel"**
		Names	name	**Meter Type (01:485,02:GPRS,04:PLC,06:RF)**
			NameType	name **"meterType"**
		Names	name	**Meter Mode (01:Postpaid,02:Prepaid)**
			NameType	name **"meterMode"**
		Names	name	**Manufacturer Code (14:HEXING,33:INHE,66:KAIFA)**
			NameType	name **"manufacturer"**
		Names	name	**Value of current transformer ratio (1/1,100/5,200/5)**
			NameType	name **"ctRadio"**
		Names	name	**Value of potential transformer ratio (1/1,100/5,200/5)**
			NameType	name **"ptRadio"**
		Names	name	**Protocol Type (03:DLMS,15:HDLC)**
			NameType	name **"protocolType"**

If MDC need other parameters, Unified System be able to extend the element which named "Names".

4.3 Examples of XML Message

4.3.1 MeterConfig

```
<?xml version="1.0" encoding="UTF-8" ?>
<RequestMessage
xmlns="http://iec.ch/TC57/2011/schema/message"
xmlns:m="http://iec.ch/TC57/2011/MeterConfig#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://iec.ch/TC57/2011/schema/message Message.xsd">
<Header>
<Verb>create</Verb>
<Noun>MeterConfig</Noun>
<Revision>2.0</Revision>
<Timestamp>2015-03-14T20:04:39+04:30</Timestamp>
<Source>Unified System</Source>
<AsyncReplyFlag>false</AsyncReplyFlag>
<AckRequired>false</AckRequired>
<User>
<UserID>...</UserID>
</User>
<MessageID>83c643e6-85c5-43c0-9e0a-fa1deb469b72</MessageID>
<CorrelationID>1001</CorrelationID>
<Property>
<Name>password</Name>
```

<Value>...</Value>
</Property>
<Property>
<Name>timeout(m)</Name>
<Value>30</Value>
</Property>
</Header>
<Request>
<m:MeterConfig>
<m:Meter>
<m:mRID>014000000011</m:mRID>
<m:Names>
<m:name>014000000011</m:name>
<m:NameType>
<m:name>serialNumber</m:name>
</m:NameType>
</m:Names>
<m:Names>
<m:name>HXEP1000T</m:name>
<m:NameType>
<m:name>meterModel</m:name>
</m:NameType>
</m:Names>
<m:Names>
<m:name>14</m:name>
<m:NameType>
<m:name>manufacturer</m:name>
</m:NameType>
</m:Names>
<m:Names>
<m:name>1/1</m:name>
<m:NameType>
<m:name>ctRadio</m:name>
</m:NameType>
</m:Names>
<m:Names>
<m:name>1/1</m:name>
<m:NameType>
<m:name>ptRadio</m:name>
</m:NameType>
</m:Names>
<m:Names>
<m:name>03</m:name>
<m:NameType>
<m:name>protocolType</m:name>
</m:NameType>
</m:Names>
</m:Meter>
</m:MeterConfig>
</Request>
</RequestMessage>

4.3.2 ReplyMeterConfig

<?xml version="1.0" encoding="UTF-8" ?>
<ResponseMessage
xmlns="http://iec.ch/TC57/2011/schema/message"
xmlns:m="http://iec.ch/TC57/2011/MeterConfig#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://iec.ch/TC57/2011/schema/message Message.xsd">
<Header>
<Verb>reply</Verb>
<Noun>MeterConfig</Noun>
<Revision>2.0</Revision>
<Timestamp>2015-06-24T02:30:00+04:30</Timestamp>
<Source>MDC-001</Source>
<User>
<UserID>...</UserID>
</User>
<MessageID>83c643e6-85c5-43c0-9e0a-fa1deb469b72</MessageID>
<CorrelationID>1001</CorrelationID>
<Property>
<Name>password</Name>
<Value>...</Value>
</Property>
</Header>
<Reply>
<Result>OK</Result>
<Error>
<code>0.0</code>
</Error>
</Reply>
</ResponseMessage>

5. DeviceArchive Synchronization

5.1 Introduction

This section aims at presenting message structures regarding adding an equipment.

5.2 Usecase Description

Firstly Unified System should be to validating the meter's information. After thatUnified System will send the command of synchronizing the divice to MDC which contains device's full profile. The details of Unified System's request to MDC are available in the following Tables:

Table 4.1: Device Archive Synchornization (add divice)

Device Addition			
Message type	Verb	Noun	Main elements
Request	create	EndDeviceConfig	Header/Payload
Response	reply	EndDeviceConfig	Header/Reply

Note:
This transaction can be conducted only in synchronous modes. In synchronous mode, a reply will sent back to Unified System, showing the successfulness/unsuccessfulness.

Table 4.2: DeviceArchive Synchornization (modify device)

Device Modification			
Message type	Verb	Noun	Main elements

Request	change	EndDeviceConfig	Header/Payload
Response	reply	EndDeviceConfig	Header/Reply

Note:
This transaction can be conducted only in synchronous modes. In synchronous mode, a reply will sent back to Unified System, showing the successfulness/unsuccessfulness.

Table 4.3: DeviceArchive Synchornization (delete device)

Device Deletion			
Message type	Verb	Noun	Main elements
Request	delete	EndDeviceConfig	Header/Payload
Response	reply	EndDeviceConfig	Header/Reply

Note:
This transaction can be conducted only in synchronous modes. In synchronous mode, a reply will sent back to Unified System, showing the successfulness/unsuccessfulness.

Payload's elements regarding this request are presented in Table 4.4:

Table 4.4: EndDeviceConfig

Payload of EndDeviceConfig					
EndDeviceConfig	EndDevcie (unbounded)	mRID	Deivce ID		
		Names	name	name	**Serial Number**
			NameType		**“serialNumber”**
		Names	name	name	**Protocol Type (16:DCU-DLMS)**
			NameType		**“protocolType”**

If MDC need other parameters, Unified System be able to extend the element which named “Names”.

5.3 Examples of XML Message

5.3.1 EndDeviceConfig

```
<?xml version="1.0" encoding="UTF-8" ?>
<RequestMessage
xmlns="http://iec.ch/TC57/2011/schema/message"
xmlns:m="http://iec.ch/TC57/2011/EndDeviceConfig#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://iec.ch/TC57/2011/schema/message Message.xsd">
<Header>
<Verb>create</Verb>
<Noun>EndDeviceConfig</Noun>
<Revision>2.0</Revision>
```

<Timestamp>2015-03-14T20:04:39+04:30</Timestamp>
<Source>MDM</Source>
<AsyncReplyFlag>false</AsyncReplyFlag>
<AckRequired>false</AckRequired>
<User>
<UserID>...</UserID>
</User>
<MessageID>83c643e6-85c5-43c0-9e0a-fa1deb469b72</MessageID>
<CorrelationID>1001</CorrelationID>
<Property>
<Name>password</Name>
<Value>...</Value>
</Property>
<Property>
<Name>timeout(m)</Name>
<Value>30</Value>
</Property>
</Header>
<Payload>
<m:EndDeviceConfig>
<m:EndDevice>
<m:mRID>014000000011</m:mRID>
<m:Names>
<m:name>014000000011</m:name>
<m:NameType>
<m:name>serialNumber</m:name>
</m:NameType>
</m:Names>
<m:Names>
<m:name>03</m:name>
<m:NameType>
<m:name>protocolType</m:name>
</m:NameType>
</m:Names>
</m:EndDevice>
</m:EndDeviceConfig>
</Payload>
</RequestMessage>

5.3.2 ReplyEndDeviceConfig

<?xml version="1.0" encoding="UTF-8" ?>
<ResponseMessage
xmlns="http://iec.ch/TC57/2011/schema/message"
xmlns:m="http://iec.ch/TC57/2011/EndDeviceConfig#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://iec.ch/TC57/2011/schema/message Message.xsd">
<Header>
<Verb>reply</Verb>
<Noun>EndDeviceConfig</Noun>
<Revision>2.0</Revision>
<Timestamp>2015-06-24T02:30:00+04:30</Timestamp>
<Source>MDC-001</Source>
<User>
<UserID>...</UserID>
</User>
<MessageID>83c643e6-85c5-43c0-9e0a-fa1deb469b72</MessageID>
<CorrelationID>1001</CorrelationID>
<Property>

<Name>password</Name>
<Value>...</Value>
</Property>
</Header>
<Reply>
<Result>OK</Result>
<Error>
<code>0.0</code>
</Error>
</Reply>
</ResponseMessage>

6. Mother-Child Meter Relation Synchronization

6.1 Introduction

According to IEC 61968-9, in order to perform a linkage between a meter and another device, "MasterDataLinkageConfig" service is used. The sketch of this service is provided in Table 5.1, Table5.2 and Table 5.3. It is noteworthy to say that, sometimes a meter is linked to some submeters.

Table 5.1: Mother-Child Meter Relation Synchronization (install child meter to mother meter)

Child Meter Installation			
Message type	Verb	Noun	Main elements
Request	create	MasterDataLinkageConfig	Header/Payload
Response	reply	MasterDataLinkageConfig	Header/Reply

Note:
This transaction can be conducted only in synchronous modes. In synchronous mode, a reply will sent back to Unified System, showing the successfulness/unsuccessfulness.

Table 5.2: Mother-Child Meter Relation Synchronization (change child meter to a new mother meter or change mother meter)

Child Meter Change			
Message type	Verb	Noun	Main elements
Request	change	MasterDataLinkageConfig	Header/Payload
Response	reply	MasterDataLinkageConfig	Header/Reply

Note:
This transaction can be conducted only in synchronous modes. In synchronous mode, a reply will sent back to Unified System, showing the successfulness/unsuccessfulness.

Table 5.3: Mother-Child Meter Relation Synchronization (remove child Meter from mother meter)

Child Meter Deletion			
Message type	Verb	Noun	Main elements
Request	delete	MasterDataLinkageConfig	Header/Payload
Response	reply	MasterDataLinkageConfig	Header/Reply

Note:
This transaction can be conducted only in synchronous modes. In synchronous mode, a reply will sent back to Unified System, showing the successfulness/unsuccessfulness.

6.2 Usecase Description

Payload's elements regarding this request are presented in Table 5.4 and 5.5:

Table 5.4: MasterDataLinkageConfig (install and remove child meter)

Payload of MasterDataLinkageConfig					
MasterDataLinkageConfig	Meter (unbounded)	mRID		Meter ID	
		Names (unbounded)	name NameType	name	**subMeter ID "subMeter"**

Table 5.5: MasterDataLinkageConfig (change mother meter)

Payload of MasterDataLinkageConfig					
MasterDataLinkageConfig	Meter (unbounded)	mRID		Meter ID	
		Names (only one)	name NameType	name	**masterMeter ID "masterMeter"**

6.3 Examples of XML message

6.3.1 MasterDataLinkageConfig

```
<?xml version="1.0" encoding="UTF-8" ?>
<RequestMessage
xmlns="http://iec.ch/TC57/2011/schema/message"
xmlns:m="http://iec.ch/TC57/2011/MasterDataLinkageConfig#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://iec.ch/TC57/2011/schema/message Message.xsd">
<Header>
<Verb>create</Verb>
<Noun>MasterDataLinkageConfig</Noun>
<Revision>2.0</Revision>
<Timestamp>2015-03-14T20:04:39+04:30</Timestamp>
<Source>Unified System</Source>
<AsyncReplyFlag>false</AsyncReplyFlag>
<AckRequired>false</AckRequired>
<User>
<UserID>...</UserID>
</User>
<MessageID>83c643e6-85c5-43c0-9e0a-fa1deb469b72</MessageID>
<CorrelationID>1001</CorrelationID>
<Property>
```

```
<Name>password</Name>
<Value>...</Value>
</Property>
<Property>
<Name>timeout(m)</Name>
<Value>30</Value>
</Property>
</Header>
<Payload>
<m:MasterDataLinkageConfig>
<m:Meter>
<m:mRID>014000000011</m:mRID>
<m:Names>
<m:name>014000000012</m:name>
<m:NameType>
<m:name>subMeter</m:name>
</m:NameType>
</m:Names>
</m:Meter>
</m:MasterDataLinkageConfig>
</Payload>
</RequestMessage>
```

6.3.2 ReplyMasterDataLinkageConfig

```
<?xml version="1.0" encoding="UTF-8" ?>
<ResponseMessage
xmlns="http://iec.ch/TC57/2011/schema/message"
xmlns:m="http://iec.ch/TC57/2011/MasterDataLinkageConfig#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://iec.ch/TC57/2011/schema/message Message.xsd">
<Header>
<Verb>reply</Verb>
<Noun>MasterDataLinkageConfig</Noun>
<Revision>2.0</Revision>
<Timestamp>2015-06-24T02:30:00+04:30</Timestamp>
<Source>MDC-001</Source>
<User>
<UserID>...</UserID>
</User>
<MessageID>83c643e6-85c5-43c0-9e0a-fa1deb469b72</MessageID>
<CorrelationID>1001</CorrelationID>
<Property>
<Name>password</Name>
<Value>...</Value>
</Property>
</Header>
<Reply>
<Result>OK</Result>
<Error>
<code>0.0</code>
</Error>
</Reply>
</ResponseMessage>
```

7. Gateway-Child Meter RelationSynchronization

7.1. Introduction

This section aims at presenting message structures regarding Installing an equipment.

7.2. Usecase Description

In order to perform a linkage between a meter and a EndDevice,
"MasterDataLinkageConfig" service is used.
The details of Unified System's request to MDC are available in the following Tables:

Table 10.1:MasterDataLinkageConfig

MasterDataLinkageConfig			
Message type	Verb	Noun	Main elements
Request	create	MasterDataLinkageConfig	Header/Payload
Response	reply	MasterDataLinkageConfig	Header/Reply

Note:
This transaction can be conducted only in synchronous modes. In synchronous mode, a reply will sent back to Unified System, showing the successfulness/unsuccessfulness.

If the accuracy of uninstallation is evident to Unified System, Unified System sends the command ofunregistering the meter to MDC.
The overview of this usecase is shown in Table below:

Table 7.2: Services used in meter uninstallation

MasterDataLinkageConfig			
Message type	Verb	Noun	Main elements
Request	delete	MasterDataLinkageConfig	Header/Payload
Response	reply	MasterDataLinkageConfig	Header/Reply

Note:
This transaction can be conducted only in synchronous modes. In synchronous mode, a reply will sent back to Unified System, showing the successfulness/unsuccessfulness.

Payload's elements regarding this request are presented in Table 7.2:

Table 7.3: **MasterDataLinkageConfig**

Payload of MasterDataLinkageConfig					
MasterDataLinkageConfig	EndDevice	mRID	Device ID		
		Names	name NameType	name	**“TermInal”** **“DeviceType”**
	Meter	mRID	Meter ID		
		Names	name NameType	name	**“Meter”** **“DeviceType”**

7.3 Examples of XML message

7.3.1 MasterDataLinkageConfig

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<RequestMessage
xmlns="http://iec.ch/TC57/2011/schema/message"
xmlns:m="http://iec.ch/TC57/2011/MasterDataLinkageConfig#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://iec.ch/TC57/2011/schema/message Message.xsd">
<Header>
<Verb>create</Verb>
<Noun>MasterDataLinkageConfig</Noun>
<Revision>2.0</Revision>
<Timestamp>2015-03-14T20:04:39+04:30</Timestamp>
<Source>Unified System</Source>
<AsyncReplyFlag>false</AsyncReplyFlag>
<AckRequired>false</AckRequired>
<User>
<UserID>...</UserID>
</User>
<MessageID>83c643e6-85c5-43c0-9e0a-fa1deb469b72</MessageID>
<CorrelationID>1001</CorrelationID>
<Property>
<Name>password</Name>
<Value>...</Value>
</Property>
<Property>
<Name>timeout(m)</Name>
<Value>30</Value>
</Property>
</Header>
<Payload>
<m:MasterDataLinkageConfig>
<m:EndDevice>
<m:mRID>01400004</m:mRID>
<m:Names>
<m:name>Meter</m:name>
<m:NameType>
<m:name>DevcieType</m:name>
</m:NameType>
</m:Names>
</m:Meter>
<m:Meter>
<m:mRID>014000000011</m:mRID>
<m:Names>
<m:name>Meter</m:name>
<m:NameType>
<m:name>DevcieType</m:name>
</m:NameType>
</m:Names>
</m:Meter>
</m:MasterDataLinkageConfig>
</Payload>
</RequestMessage>
```

7.3.2 ReplyMasterDataLinkageConfig

```
<?xml version="1.0" encoding="UTF-8" ?>
<ResponseMessage
xmlns="http://iec.ch/TC57/2011/schema/message"
xmlns:m="http://iec.ch/TC57/2011/MasterDataLinkageConfig#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://iec.ch/TC57/2011/schema/message Message.xsd">
```

```
<Header>
<Verb>reply</Verb>
<Noun>MasterDataLinkageConfig</Noun>
<Revision>2.0</Revision>
<Timestamp>2015-06-24T02:30:00+04:30</Timestamp>
<Source>MDC-001</Source>
<User>
<UserID>...</UserID>
</User>
<MessageID>83c643e6-85c5-43c0-9e0a-fa1deb469b72</MessageID>
<CorrelationID>1001</CorrelationID>
<Property>
<Name>password</Name>
<Value>...</Value>
</Property>
</Header>
<Reply>
<Result>OK</Result>
<Error>
<code>0.0</code>
</Error>
</Reply>
</ResponseMessage>
```

8. Token Sending

8.1 Introduction

Some times we need intergrate for Prepay system, and it should have the ability to send Tokens to meters.

This section aims at presenting message structure regarding this usecase. General Usecase specifications along with the services used in this respect are provided in Table 5.1.

Table 6.1: Services used in Sending Tokens to Interface Billing

Sending Tokens			
Type of message	Verb	Noun	Mainelements
Request*	create	EndDeviceControls	Header/Payload
Acknowledgement	reply	EndDeviceControls	Header/Reply
Response**	reply	EndDeviceControls	Header/Reply

Note:

This transaction can be conducted only in asynchronous modes.

*The token is stored in payload's name node in sequence, the data format: token1,token2,token3…

** This response used to Indicate the successfulness/unsuccessfulness of performing the command on the meter in asynchronous transaction.The execution result of token is stored in reply's detail node, the data format:token1:result1,token2:result2…, the result code reference on table 6.2

Table 6.2: Result Code of Token

Code	Name
0	Token Processing Successful
1	Token Interpretation Error
2	Token Used
3	Token Expired
4	Key Expired
5	Recharge Value Exceed the Accumulate Amount Limit
6	Key Type Don't Allow Recharge
7	Test Token Generated by Non-specified Manufacturer
8	If input the 1st modifed key Token, then input is not the 2nd modified key TOKEN
9	Key Type shouldn't change to 3
10	If the parent key type is not the initial one, the child key type can't be changed to initial key
11	Token serial number not correct
12	Token setting parameter invalid

8.2 Usecase Description

Table6.3 shows payload's elements for Prepaycommand to MDC regarding this usecase:

Table 6.3: Payload for EndDeviceControls (Sending Tokens)

EndDeviceControls	EndDeviceControl(Unbounded)	reason		**“sendTokens”**	
		EndDeviceControlType ref		15.13.112.78	
		EndDevices	mRID	Meter ID	
			Names	name	Token String

In this case, it provide two ways for bulking operations, one is you can set multily tokens for a meter, and the other one is you can set multily tokens for multily meters when the meters use the same tokens.

8.3 Examples of XML Message

8.3.1 SendTokens

```
<?xml version="1.0" encoding="utf-8"?>
<RequestMessage
xmlns="http://iec.ch/TC57/2011/schema/message"
xmlns:m="http://iec.ch/TC57/2011/EndDeviceControls#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://iec.ch/TC57/2011/schema/message Message.xsd">
<Header>
<Verb>create</Verb>
<Noun>EndDeviceControls</Noun>
<Revision>2.0</Revision>
<Timestamp>2016-01-01T00:00:00+04:30</Timestamp>
<Source>Unified System</Source>
<AsyncReplyFlag>true</AsyncReplyFlag>
<ReplyAddress>http://ip:port/AmiWeb/services/Metering</ReplyAddress>
<AckRequired>true</AckRequired>
<User>
<UserID>...</UserID>
</User>
<MessageID>83c643e6-85c5-43c0-9e0a-fa1deb469b72</MessageID>
```

<CorrelationID>1001</CorrelationID>
<Property>
<Name>password</Name>
<Value>...</Value>
</Property>
<Property>
<Name>timeout(m)</Name>
<Value>30</Value>
</Property>
</Header>
<Payload>
<m:EndDeviceControls>
<m:EndDeviceControl>
<m:reason>SendTokens</m:reason>
<m:EndDeviceControlType ref="15.13.112.78"/>
<m:EndDevices>
<m:mRID>014000000012</m:mRID>
<Names>
<name>Token1,Token2,Token3</name>
</Names>
</m:EndDevices>
</m:EndDeviceControl>
</m:EndDeviceControls>
</Payload>
</RequestMessage>

8.3.2 Acknowledgement

<?xml version="1.0" encoding="UTF-8"?>
<ResponseMessage
xmlns= "http://iec.ch/TC57/2011/schema/message"
xmlns:xsi= "http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation= "http://iec.ch/TC57/2011/schema/message Message.xsd">
<Header>
<Verb>reply</Verb>
<Noun>EndDeviceControls</Noun>
<Revision>2.0</Revision>
<Timestamp>2015-06-24T02:30:00+04:30</Timestamp>
<Source>MDC-001</Source>
<User>
<UserID>...</UserID>
</User>
<MessageID>83c643e6-85c5-43c0-9e0a-fa1deb469b72</MessageID>
<CorrelationID>1001</CorrelationID>
<Property>
<Name>password</Name>
<Value>...</Value>
</Property>
</Header>
<Reply>
<Result>OK</Result>
<Error>
<code>0.3</code>
</Error>
</Reply>
</ResponseMessage>

8.3.3 ReplySendTokens

<?xml version="1.0" encoding="UTF-8" ?>
<ResponseMessage

```
xmlns="http://iec.ch/TC57/2011/schema/message"
xmlns:m="http://iec.ch/TC57/2011/EndDeviceControls#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://iec.ch/TC57/2011/schema/message Message.xsd">
<Header>
<Verb>reply</Verb>
<Noun>EndDeviceControls</Noun>
<Revision>2.0</Revision>
<Timestamp>2015-03-14T20:04:39+04:30</Timestamp>
<Source>MDC-001</Source>
<User>
<UserID>...</UserID>
</User>
<MessageID>601BD4C0-F3E2-4833-BF83-7494460E74BE</MessageID>
<CorrelationID>1001</CorrelationID>
<Property>
<Name>password</Name>
<Value>...</Value>
</Property>
<Property>
<Name>timeout(m)</Name>
<Value>1</Value>
</Property>
</Header>
<Reply>
<Result>OK</Result>
<Error>
<code>0.0</code>
<details>token1:0,token2:1,token3:2</details>
</Error>
</Reply>
```

9. On Demand Meter Reading

9.1 Introduction

This section aims at presenting message structures regarding the second usecase of MDC. In this usecase, the process of on demand meter reads is clarified in detail. Table 7.1 provides the general message structure of this usecase:

Table 7.1: Meter Reading

Meter Reading			
Type of message	Verb	Noun	Mainelements
Request	Get	MeterReadings	Header/Payload
Acknowledgement	Reply	MeterReadings	Header/Reply
Response	Reply	MeterReadings	Header/Reply/Payload

9.2 Usecase Description

Request elements regarding the on demand meter reading is depicted in Table 7.2. In this usecase, for demanding meter readings, the OBIS of DLMS should be included the request message which are available in Table 7.4 of this document.

Table 7.2: Request of GetMeterReadings

GetMeter Readings	EndDevice (Unbounded)	mRID			**MeterID**
	ReadingType (Unbounded)	Names	name		OBIS ofDLMS
			NameType	**name**	"ReadingType"

Payload of the response message can be observed in Table 4.3.

Table 4.3: Payload of MeterReading

MeterReadings	MeterReading (unbounded) Note:Each meterhasitsown" MeterReading"	Meter	**mRID**	MeterID
		Readings (unbounded) Note: For each reading interval, one "Readings" is considered. Moreover, for each "Readings" there is a "ReadingType Ref"	reason	"inquiry"
			timeStamp	
			value	Readvalue
			ReadingTyperef	OBIS of DLMS

The ReadingTypes about instantaneousdata reference on the blue document.

9.3 Examples of XML Message

9.3.1 MeterReadings

```
<?xml version="1.0" encoding="UTF-8" ?>
<RequestMessage
xmlns="http://iec.ch/TC57/2011/schema/message"
xmlns:m="http://iec.ch/TC57/2011/GetMeterReadings#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://iec.ch/TC57/2011/schema/message Message.xsd">
<Header>
<Verb>get</Verb>
<Noun>MeterReadings</Noun>
<Revision>2.0</Revision>
<Timestamp>2015-03-14T20:04:39+04:30</Timestamp>
<Source>Unified System</Source>
<AsyncReplyFlag>true</AsyncReplyFlag>
<ReplyAddress>http://ip:port/AmiWeb/services/Metering</ReplyAddress>
<AckRequired>true</AckRequired>
<User>
<UserID>...</UserID>
</User>
<MessageID>83c643e6-85c5-43c0-9e0a-fa1deb469b72</MessageID>
<CorrelationID>1001</CorrelationID>
<Property>
<Name>password</Name>
<Value>...</Value>
</Property>
```

```
<Property>
<Name>timeout(m)</Name>
<Value>1</Value>
</Property>
</Header>
<Request>
<m:GetMeterReadings>
<m:EndDevice>
<m:mRID>014000000012</m:mRID>
</m:EndDevice>
<m:ReadingType>
<m:Names>
<m:name>3#1.0.31.7.0.255#2</m:name>
<m:NameType>
<m:name>ReadingType</m:name>
</m:NameType>
</m:Names>
</m:ReadingType>
</m:GetMeterReadings>
</Request>
</RequestMessage>
```

9.3.2 Acknowledgement

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseMessage
xmlns= "http://iec.ch/TC57/2011/schema/message"
xmlns:xsi= "http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation= "http://iec.ch/TC57/2011/schema/message Message.xsd">
<Header>
<Verb>reply</Verb>
<Noun>MeterReadings</Noun>
<Revision>2.0</Revision>
<Timestamp>2015-06-24T02:30:00+04:30</Timestamp>
<Source>MDC-001</Source>
<User>
<UserID>...</UserID>
</User>
<MessageID>83c643e6-85c5-43c0-9e0a-fa1deb469b72</MessageID>
<CorrelationID>1001</CorrelationID>
<Property>
<Name>password</Name>
<Value>...</Value>
</Property>
</Header>
<Reply>
<Result>OK</Result>
<Error>
<code>0.3</code>
</Error>
</Reply>
</ResponseMessage>
```

9.3.3 ReplyMeterReading

```
<?xml version="1.0" encoding="UTF-8" ?>
<ResponseMessagexmlns="http://iec.ch/TC57/2011/schema/message"
xmlns:gmr="http://iec.ch/TC57/2011/GetMeterReadings#"
xmlns:m="http://iec.ch/TC57/2011/MeterReadings#"
```

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://iec.ch/TC57/2011/schema/message Message.xsd">
<Header>
<Verb>reply</Verb>
<Noun>MeterReadings</Noun>
<Revision>2.0</Revision>
<Timestamp>2015-03-14T20:05:08+04:30</Timestamp>
<Source>MDC-001</Source>
<User>
<UserID>...</UserID>
</User>
<MessageID>601BD4C0-F3E2-4833-BF83-7494460E74BE</MessageID>
<CorrelationID>1001</CorrelationID>
<Property>
<Name>password</Name>
<Value>...</Value>
</Property>
<Property>
<Name>timeout(m)</Name>
<Value>1</Value>
</Property>
</Header>
<Reply>
<Result>OK</Result>
<Error>
<code>0.0</code>
</Error>
</Reply>
<Payload>
<m:MeterReadings>
<m:MeterReading>
<m:Meter>
<m:mRID>014000000012</m:mRID>
</m:Meter>
<m:Readings>
<m:reason>inquiry</m:reason>
<m:timeStamp>2015-03-14T20:05:03+04:30</m:timeStamp>
<m:value>141</m:value>
<m:ReadingType ref="3#1.0.31.7.0.255#2"/>
</m:Readings>
</m:MeterReading>
</m:MeterReadings>
</Payload>
</ResponseMessage>

10. Meter Connect and Disconnect

10.1 Introduction

The current use case describes the process of remote disconnection or reconnection of electricity supply to a customer on a designated date.

The steps regarding this use case (disconnection) are:

1. The EndDeviceControl message is sent from the Unified System to the MDC to disconnect the meter.
2. Remote meter disconnect is performed.

General Usecase specifications along with the services used in this respect are provided in Table 8.1.

Table 8.1: Services used for Meter Disconnect and Reconnect

Disconnect					
Message type	**Verb**	**Noun**	**Main elements**	**Sender**	**Receiver**
Request	create	EndDeviceControls	Header/ Payload	Unified System	MDC
Acknowledgement	reply	EndDeviceControls	Header/Reply	MDC	Unified System
Response	reply	EndDeviceControls	Header/Reply	MDC	Unified System

Note that following the disconnection, there may be ameter reconnect sequence which is performed as below:

1. The EndDeviceControl message is sent from the Unified System to the MDC to reconnect the meter.
2. Remote meter reconnect is performed.

10.2 Usecase Description

Table 8.3 shows payload's elements for EndDeviceControls message:

Table 8.3: Request of EndDeviceControls

Payload of EndDeviceControls						
EndDeviceControls	EndDeviceControl(Unbounded)	reason		"Disconnect"/"Reconnect"		
		EndDeviceControlType ref		Control Code(refer to Table 8.5)		
		EndDevices(Unbounded)	mRID	Meter ID		
			Names	name NameType	"Disconnect"/"Reconnect" name	"ControlType"

Details of the Simple acknowledgement from MDC to Unified System in the case of asynchronous transaction are provided in Table 2.5.

More details regarding EndDeviceControlType are provided in Table 8.5.

Table 8.5: Controlling code

ControlDescription	Control Code	Type	Domain	SubDomain	Action
Remote disconnection	3.0.211.23	ElectricMeter	n/a	RemoteAccess	Disconnect
Remote connection	3.0.211.18	ElectricMeter	n/a	RemoteAccess	connect

10.3 Examples of XML Message

10.3.1 MeterConenctDisconnect

```
<?xml version="1.0" encoding="utf-8"?>
<RequestMessage
xmlns="http://iec.ch/TC57/2011/schema/message"
xmlns:m="http://iec.ch/TC57/2011/EndDeviceControls#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://iec.ch/TC57/2011/schema/message Message.xsd">
<Header>
<Verb>create</Verb>
<Noun>EndDeviceControls</Noun>
```

<Revision>2.0</Revision>
<Timestamp>2016-01-01T00:00:00+04:30</Timestamp>
<Source>Unified System</Source>
<AsyncReplyFlag>true</AsyncReplyFlag>
<ReplyAddress>http://ip:port/AmiWeb/services/Metering</ReplyAddress>
<AckRequired>true</AckRequired>
<User>
<UserID>...</UserID>
</User>
<MessageID>83c643e6-85c5-43c0-9e0a-fa1deb469b72</MessageID>
<CorrelationID>1001</CorrelationID>
<Property>
<Name>password</Name>
<Value>...</Value>
</Property>
<Property>
<Name>timeout(m)</Name>
<Value>30</Value>
</Property>
</Header>
<Payload>
<m:EndDeviceControls>
<m:EndDeviceControl>
<m:reason>Disconnect/Reconnect</m:reason>
<m:EndDeviceControlType ref="3.0.211.23"/>
<m:EndDevices>
<m:mRID>014000000012</m:mRID>
<m:Names>
<m:name>Disconnect</m:name>
<m:NameType>
<m:name>ControlType</m:name>
</m:NameType>
</m:Names>
</m:EndDevices>
</m:EndDeviceControl>
</m:EndDeviceControls>
</Payload>
</RequestMessage>

10.3.2 Acknowledgement

<?xml version="1.0" encoding="UTF-8"?>
<ResponseMessage
xmlns= "http://iec.ch/TC57/2011/schema/message"
xmlns:xsi= "http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation= "http://iec.ch/TC57/2011/schema/message Message.xsd">
<Header>
<Verb>reply</Verb>
<Noun>EndDeviceControls</Noun>
<Revision>2.0</Revision>
<Timestamp>2015-06-24T02:30:00+04:30</Timestamp>
<Source>MDC-001</Source>
<User>
<UserID>...</UserID>
</User>
<MessageID>83c643e6-85c5-43c0-9e0a-fa1deb469b72</MessageID>
<CorrelationID>1001</CorrelationID>
<Property>

<Name>password</Name>
<Value>...</Value>
</Property>
</Header>
<Reply>
<Result>OK</Result>
<Error>
<code>0.3</code>
</Error>
</Reply>
</ResponseMessage>

10.3.3 ReplyMeterConnectDisconnect

<?xml version="1.0" encoding="UTF-8" ?>
<ResponseMessage
xmlns="http://iec.ch/TC57/2011/schema/message"
xmlns:m="http://iec.ch/TC57/2011/EndDeviceControls#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://iec.ch/TC57/2011/schema/message Message.xsd">
<Header>
<Verb>reply</Verb>
<Noun>EndDeviceControls</Noun>
<Revision>2.0</Revision>
<Timestamp>2015-04-25T11:11:03+04:30</Timestamp>
<Source>MDC-001</Source>
<User>
<UserID>...</UserID>
</User>
<MessageID>601BD4C0-F3E2-4833-BF83-7494460E74BE</MessageID>
<CorrelationID>1001</CorrelationID>
<Property>
<Name>password</Name>
<Value>...</Value>
</Property>
<Property>
<Name>timeout(m)</Name>
<Value>1</Value>
</Property>
</Header>
<Reply>
<Result>OK</Result>
<Error>
<code>0.0</code>
</Error>
</Reply>
</ResponseMessage>

11. Meter Reading Data

Daily frozen data, monthly fronzen data and event data will transferred by file from MDC to Unified System.

11.1. File Transfer Rule

1) The file interface interacts via FTP to transfer data, the file is in CSV format.
2) FTP set up two dedicated accounts for Unified System and MDC, limit access to client IP to ensure security.
3) FTP has two directories: one is “exportDir” which store the data file exported by MDC, another directory is “historyDir” which store historical data, the data file will move to “historyDir” which has been processed by Unified System, historical data store in the folder named: yyyyMMdd.
4) Naming rules of data file: the data fileis divided into three types: daily frozen data(01_yyyyMMddHHmmss.csv), monthly frozen data(02_yyyyMMddHHmmss.csv) and event data(03_yyyyMMddHHmmss.csv).
5) In consideration ofdataprocessing efficiency, each data file contain maximum 100000 pieces of data.
6) The first line of data file is the name of each data field.
7) Data read and write synchronization: when exporting data, MDC produce alock file with the suffix .lock which has the same name with the data file, the lock file will be deleted after MDCcomplete exporting the data file.
8) Data exporting frequency, daily frozen data and monthly frozen data: once a day, event data: once one hour.

11.2. Data Format

11.2.1. Daily Frozen Data

Column	Data Name	Data Type	Description
1	Meter No.	String(12)	sb_dlsj.sjid
2	Data Time	String(8)	sb_dlsj.sjsj,yyyyMMdd
3	Active Energy(+) Total	NUMBER(16,4)	sb_dlsj.zxygz,kWh
4	Active Energy(+) T1	NUMBER(16,4)	sb_dlsj.zxygz1,kWh
5	Active Energy(+) T2	NUMBER(16,4)	sb_dlsj. zxygz2,kWh
6	Active Energy(+) T3	NUMBER(16,4)	sb_dlsj. zxygz3,kWh
7	Active Energy(+) T4	NUMBER(16,4)	sb_dlsj. zxygz4,kWh
12	Reactive Energy(+) Total	NUMBER(16,4)	sb_dlsj.zxwgz,kvarh
13	Meter Balance	NUMBER(16,4)	sb_dlsj.dbye,TK

11.2.2. Monthly Frozen Data

Column	Data Name	Data Type	Description
1	Meter No.	String(12)	sb_dlsj_ydj.sjid
2	Data Time	String(6)	sb_dlsj_ydj.sjsj,yyyyMM
3	Active Energy(+) Total	NUMBER(16,4)	sb_dlsj_ydj.zxygz,kWh
4	Active Energy(+) T1	NUMBER(16,4)	sb_dlsj_ydj.zxygz1,kWh
5	Active Energy(+) T2	NUMBER(16,4)	sb_dlsj_ydj.zxygz2,kWh
6	Active Energy(+) T3	NUMBER(16,4)	sb_dlsj_ydj.zxygz3,kWh
7	Active Energy(+) T4	NUMBER(16,4)	sb_dlsj_ydj.zxygz4,kWh
12	Reactive Energy(+) Total	NUMBER(16,4)	sb_dlsj_ydj.zxwgz,kvarh
13	Meter Balance	NUMBER(16,4)	sb_dlsj_ydj.dbye,TK

11.2.3. Meter Event

Column	Data Name	Data Type	Description
1	Meter No.	String(12)	sb_gj.sjid
2	Event Code	String(30)	sb_gj.gjbm
3	Event Start Time	String(14)	sb_gj.rqsj,yyyyMMddHHmmss
4	Event End Time	String(14)	sb_gj.jssj,yyyyMMddHHmmss

[N.B. If any changes made in API that will be disclosed to the tenderer before presentation/ demonstration of the functionality on the sample meters and pre-payment metering system.]