Landis+Gyr AG
Alte Steinhauserstrasse 18
6330 Cham
Switzerland

Cham, February 17, 2021 **| CONFIDENTIAL**

CONSUMER INFORMATION INTERFACE (CII) SPECIFICATION

Only to be shared upon Landis+Gyr AG approval

1 | Table of Content

1.1 | Overview

One of major drivers for the introduction of smart metering is to provide the consumer with suitable metering information to optimise his energy consumption and/ or production. For that purpose, smart meters should be equipped with local interfaces transmitting metering data towards the consumer. The requirements on the physical type of the interface, the choice of data transmitted and the transmitting schedule is highly depending on the markets and projects the meter is designed for. Concerning that the complete implementation should be done well structured and layered to provide sufficient flexibility to fulfil previously addressed requirements.

The main capabilities of the CII interface are:

- Data is transmitted in one direction only – from the DLMS/COSEM server (e.g. meter) outwards; i.e. the CII does not accept any communication data from the outside (no interference with the meter possible).
- Any subset of the data available in the meter can be transmitted via this interface.
- The selection of the subset is configurable by authorized parties (Clients).
- Cryptographic protection ensures privacy and authenticity of the data provided at the interface.
- The data transmission may be time or event triggered. The refresh rate of the data must be short enough to support time-critical applications.

The data transmission from the local interface to the consumer is not in the scope of this specification.

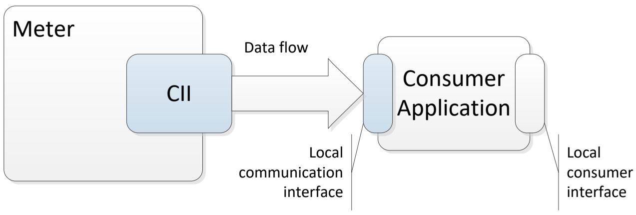


Figure 1 Basic concept

Figure 2 shows the common reference model used for the local data communication profiles. It is based on the collapsed three-layer architecture typically used in the IEC 62056 profiles (optical port, M-Bus). The Application Process, the application layer and the data link layer are specified in the IEC 62056 standards referenced in Figure 4. HDLC is the default link layer used with two physical media specific layers, optical port (IEC 62056) and wired M-Bus (EN 13575-2).

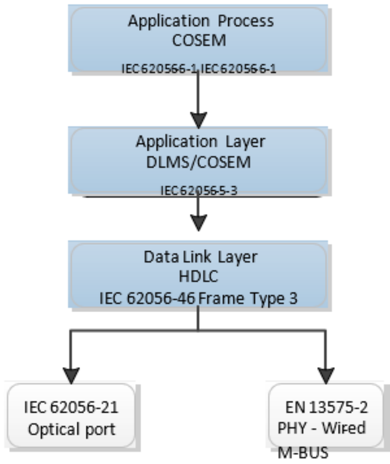


Figure 2 Communication reference model

1.2 | Application Process

The data collected by smart meter can be presented to the customer via in-home-unit. The data flow is unidirectional, always from the smart meter to the in-home-unit.
To model the data exchange between a meter and an in-home-unit the client server model is used. The meter has a role of DLMS/COSEM server (physical device) which runs an application process represented by a logical device. The in-home-unit device has a role of DLMS/COSEM client which uses the services of the application process (logical device) running in the meter.

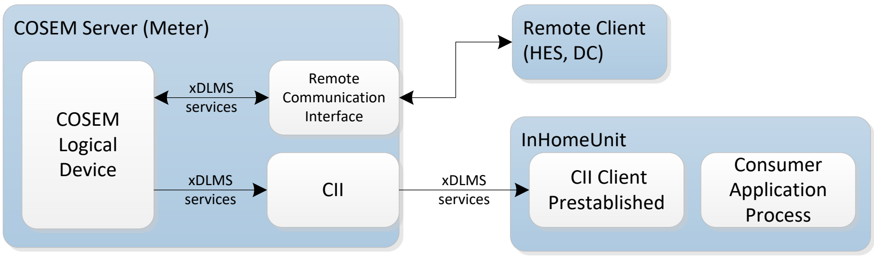


Figure 3 Application reference model

The physical and logical parameters configuring media specific part of CII, the data to be transferred and scheduling of CII communication are modelled as instances of the corresponding interface classes (objects) in the logical device (meter). These objects are managed by a remote DLMS/COSEM client running on HES, Data concentrator or any kind of remote reading tool, Figure 3.
The client running in the in-home-unit uses a dedicated address [103] and operates in a pre-established association with a COSEM Logical device running in a meter. Push is from server [1] to CII client [103] using data notification service.

1.3 | Object model

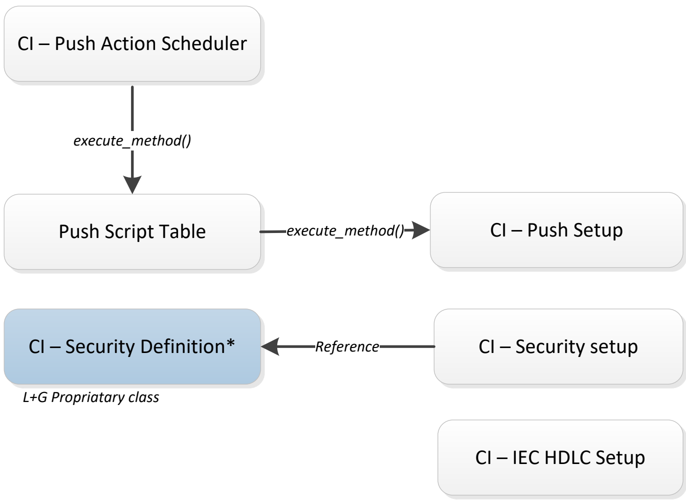


Figure 4 COSEM Interfaces classes

The objects used to configure and manage activity of the CII interfaces can be divided in the following groups:

- Scheduling and the data selection
  - At least one instance of the class “Push setup”, version 0 (class_id 40) to define a sending destination and message type.

- At least one instance of the class “Single action schedule”, version (class_id) to trigger the data transmission between the meter (server) and home display (client) synchronously with the meter clock. The data transmission could be also triggered asynchronously by an internal event e.g. alarm, receiving message form HES, etc.
- One instance of the class “Script Table”, (class-id 9, version 0) to enable triggering mechanism by “Single action schedule” class.

- Communication profile related
  - One instance of the “IEC HDLC Setup” class, (class_id 23, version 1) for setting up the data exchange over the suitable local port.
- Security definition and setup
  - One instance of the class “Security definition”, version 0 (class_id [30035-0-0-0]) defining pre-established CII server
  - One instance of the “Security setup” class, (class_id 64, version 0) to define security levels used with the message transportation.

1.3.1 | Scheduling and data selection

The core element of the CII functionality is “Data push mechanism” which is built around the “Push setup” class, which contains a list of references to COSEM objects attributes to be transmitted to the CII client (home display), push destination and communication profile used by CII.
The values of the COSEM object attributes referenced by push_object_list attribute of the Push setup class are sent to a dedicated destination whenever the push_method of the Push setup class is invoked. The method invocation could be triggered by an instance of Single Action Schedule class or internal event.
The usage of Single Action Schedule gives possibility to define the periodical or non-periodical triggers for pushing the data to client. To have possibility to use both kinds of triggers it should be possible to configure all Each Single Action Schedule instance used by CII functionality as type 5. The “basic” periodical triggers like:

- every second,
- every minute,
- hourly,
- daily,
- weekly,
- mothly,
- yearly

could be already configured by using wildcards in date and time fields of attribute execution_time_date.

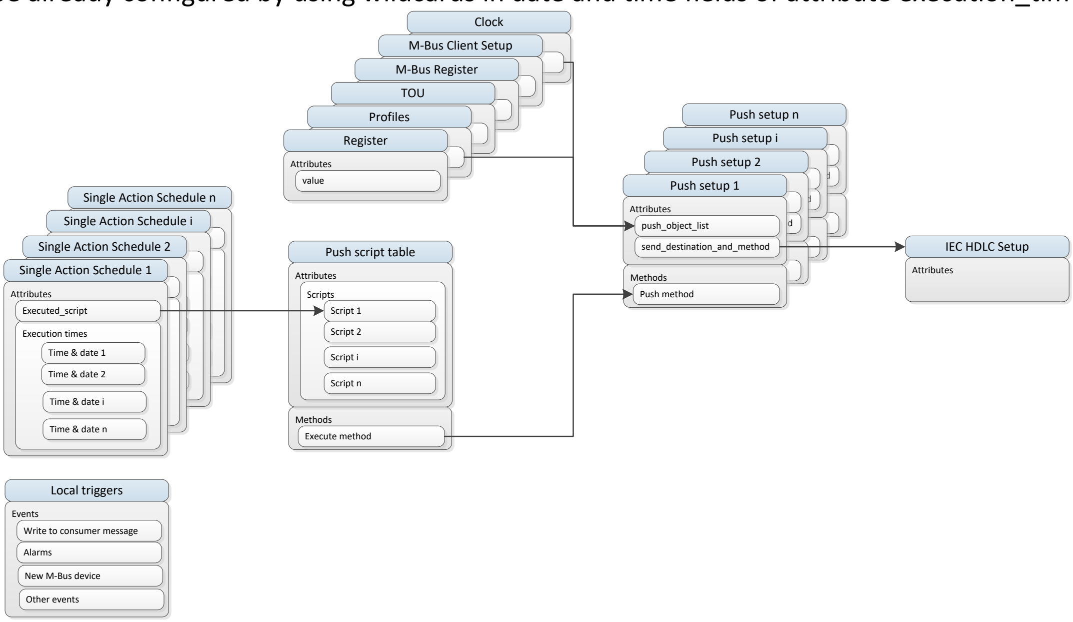


Figure 5 COSEM Interfaces classes modelling CII push operation

To get greater flexibility and possibility to specify the periods which are in between two “basic periods” the execution_time_date attribute should be an array of size between 0 and 12. With specifying 12 execution_time_date is possible to divide each period defined by wildcards to 12 subintervals:

- 1, 5, 10, 15, 30, 60 seconds
- 1, 5, 10, 15, 30, 60 minutes
- 1, 2, 4, 6, 8, 12, 24 hours
- Weekly
- Monthly
- Yearly

	Entry	Date				Time			
Period		**Y**	**M**	**D**	**WD**	**H**	**m**	**s**	**hs**
Not specified	1	FFFF	FF	FF	FF	FF	FF	FF	FF
Every second	1	FFFF	FF	FF	FF	FF	FF	FF	00
Every minute at 0 seconds	1	FFFF	FF	FF	FF	FF	FF	00	00
Every minute at 8 seconds	1	FFFF	FF	FF	FF	FF	FF	08	00
Every minute at 24 seconds	1	FFFF	FF	FF	FF	FF	FF	18	00
Every 5 second	1	FFFF	FF	FF	FF	FF	FF	00	00
	2	FFFF	FF	FF	FF	FF	FF	05	00
	3	FFFF	FF	FF	FF	FF	FF	0A	00
	4	FFFF	FF	FF	FF	FF	FF	0F	00
	5	FFFF	FF	FF	FF	FF	FF	14	00
	6	FFFF	FF	FF	FF	FF	FF	19	00
	7	FFFF	FF	FF	FF	FF	FF	1D	00
	8	FFFF	FF	FF	FF	FF	FF	23	00
	9	FFFF	FF	FF	FF	FF	FF	28	00
	10	FFFF	FF	FF	FF	FF	FF	2D	00
	11	FFFF	FF	FF	FF	FF	FF	32	00
	12	FFFF	FF	FF	FF	FF	FF	37	00
Every 15 seconds	1	FFFF	FF	FF	FF	FF	FF	00	00
	2	FFFF	FF	FF	FF	FF	FF	0F	00
	3	FFFF	FF	FF	FF	FF	FF	1D	00
	4	FFFF	FF	FF	FF	FF	FF	2D	00

Table 1 Examples of triggering periods

Legend for Table 1:	
Y - year	*h - hours*
M - month	*m - minutes*
D - day	*s - seconds*
WD - week day	*hs - hundreds of milliseconds*
Entry – entry index in execution_time_date attribute	
The numbers for the date and time fields are in HEX notation	

In Table 1 are given examples of few possible configurations of execution_time_date attribute.The Every 5 second and Every 15 seconds examples show, how the shorter periods than 1 minute could be configured by using different number of entries in execution_time_date attribute. In case if there are multiple entries specifying very same time then the script referenced in the second attribute of the single action schedule is executed just once.
Whenever the attribute is configured the complete attribute (array) is written by single write command which could contain the numbers of entries in range 0 to maximum size (12). If a client tries to write an array to execution_time_date attribute which is larger than the attribute size the server (meter) reports TYPE_UNMATCHED error. Once the data are received a server checks the validity of the *date* and *time* fields for each entry. The values should be in valid ranges for *date* and *time* according to DLMS-UA 1000-1 Ed. 11.0. Only exception is usage of hundredths of second where allowed values are 00 and FF, see Table 2. If the value in any of entries is out of range the write for complete array is discarded and server issues error OTHER_REASON.

Time structure	Range
hours	0..59, $(FF)_{hex}$
Minutes	0..59, $(FF)_{hex}$
Seconds	0..59, $(FF)_{hex}$
hundredths	0, $(FF)_{hex}$

Table 2 Ranges of values in time structure

Even when the all values in date and time fields are valid an additional validation of wildcards usage in time field is taken. Whenever any of the fields hours, minutes or seconds is specified the fields following that field should not contain wildcard, $(FF)_{hex}$. The valid combinations of the specified values and wildcard are presented in Table 3.

Time			
hours	**minutes**	**seconds**	**Hundredths**
FF	FF	FF	FF
FF	FF	FF	xx
FF	FF	xx	xx
FF	FF	xx	xx
FF	xx	xx	xx
xx	xx	xx	xx

Table 3 Usage of wildcards in time structure, Xx =[00..59]

The Single Action Schedules should follow implementation according DLMS-UA 1000-1 Ed. 11.0.
As long as the HDLC communication profile is selected and used independently of the physical layer, the transport_service_type and message_type fields of the send_destination_and_method should reference HDLC and A-XDR encoded COSEM APDU respectively. The destination field of the previously mentioned attribute contains the reference to the instance of the HDLC Setup class ((OBIS code) which provides configuration of communication port used for CII.

1.3.2 | Security Definition

Security definition – CII Server is new Security definition to define Consumer information interface. Refer [09.04.01.e]_Data_Security for details. CII client server pre-established association, current association (0-0:40.0.0.255) does not reflect association state of CII. Most of the association parameters can be read from Security definition – CII server objects. CII will have its own security context and keys. At present CII supports only LN association. It can be extended later to support both LN and SN pre-established associations.
Security definition object is used to model CII server, and it sends data out, maybe in future it can be used as bidirectional communication
The Application process is defined by Scheduler, Script table and Push object. The application process is linked to physical channel by use of attribute send_destination_and_method_type of push setup object.

```
send_destination_and_method ::= structure
{
transport_service: transport_service_type,
destination: octet-string,
message: message_type
}
```

For CII,
Transport service type supported is HDLC. Destination is reference of object used to send HDLC data to designated physical port, supported destinations for CII are

1. IEC HDLC setup - Consumer Information Interface 1: this is default Interface supported by CII over optical interface. Supported baud rate 300 .. 9600 with 9600 as default baud rate.

2. IEC HDLC setup - Consumer Information Interface 2: This is additional Interface which can be used to configure CII over wired M-Bus interface. The L2 layer in this case would be HDLC (frame type 3) and physical layer would be same wire medium as used by device to communicate to other M-Bus devices over this Meter Bus. Supported baud rate is 300 .. 9600 with 2400 as default baud rate.

Possible future extensions: It is possible to support CII over common IP interface with TCP/UDP as transport service.
Only message type supported is (0) A-XDR encoded xDLMS APDU.

1.3.3 | Security

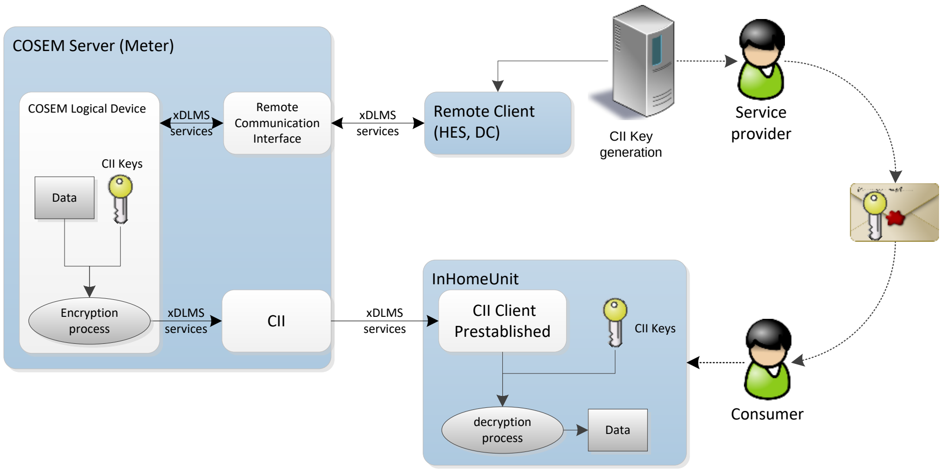


Figure 6 CII Security management

The data pushed from the application process to the CII may be secured (encryption and/or authentication) by the meter. The level of security is configured in CII specific instance of the COSEM Security setup class (class_ID 64) named “Security setup-Consumer Information” (logical_name 0-0:43.0.1.255).
• If it is secured, then security suite 0 is applied.
• The security material used for this Meter-CII- ConsumerEquipment communication is independent of the security material used for the remote Meter-HES communication.

The CIP security context is defined in a dedicated security setup object.
The keys (CIP keys) used for the data pushed to the CII are managed by the HES (comp. Figure 6) . To change a CIP key:
1. the HES wraps the new CIP key with the meter’s master key,
2. the HES sends the wrapped key to the meter using the method global_key_transfer of the object “Security setup-Consumer Information” (logical_name: 0-0:43.0.1.255) via the Management Client association.
The delivery of the appropriate key to the consumer equipment (user of the CII) is out of scope of this document.
The CIP frame counters used for the data pushed to the CII are not accessible from outside the meter (neither by the HES nor by the consumer equipment). Upon reception of a new set of keys from the HES the meter resets the CIP frame counters .The meter increments the frame counter with every message pushed.

Conclusion: the CIP data can be send as:

0: non-ciphered push message
1: authenticated (push message will have an authentication tag) -> authentication key will be used
2: encrypted (push message will be encrypted) -> encryption key will be used
3: authenticated encryption (push message will be encrypted and an authentication tag will be generated)
-> both authentication and encryption key will be used.

1.4 | Application Layer

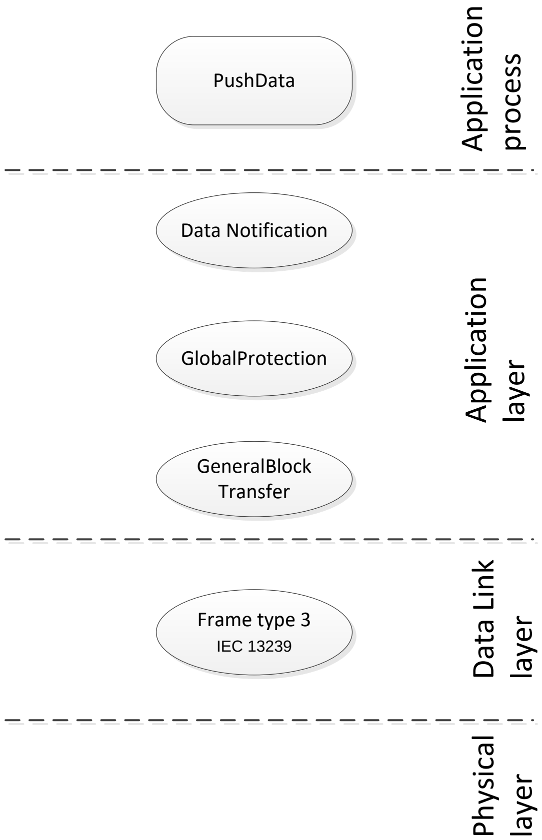


Figure 7 Data flow

Services: Check in standard

- DataNotification
- GeneralProtection
- GeneralBlockTransver

Explain type of general block transfer and Aborting it
General-Glo-Ciphering ::= SEQUENCE
{
system-title OCTET STRING,
ciphered-content OCTET STRING
}
Security control byte

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3..0
Compression	Key_Set	E	A	Security_Suite_Id

The Key_Set bit is not relevant and shall be set to 0 when the service specific dedicated ciphering is used.

1.5 | Data Link Layer
1.5.1 | HDLC

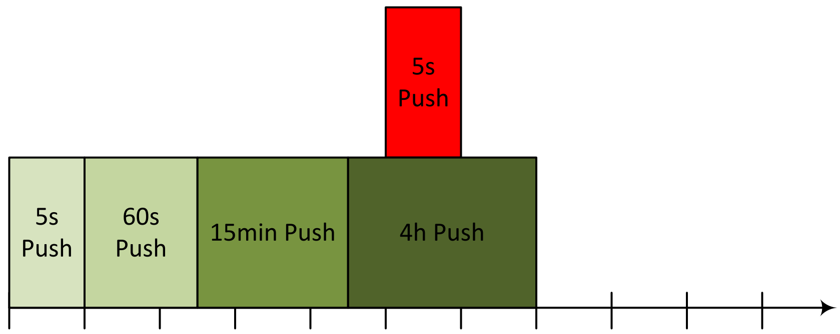
- Frame type 3 and the non-basic frame format transparency according to IEC 13239, sect. 4.3.3 is used.
- The meter acts as HDLC primary/control station according to IEC 13239 (sect. 6.13 Unbalanced connectionless operation).
- The control station sends unsolicited UI frames carrying the data as configured in the PUSH setup.
- According to IEC 13239 sect. 6.13.4.2.1: whenever the control station is ready to send a UI command frame, it shall send it immediately since there is no flow control in connectionless class service. The tributary station(s) shall only send UI response frames when given permission to do so.
- The connection is unidirectional from meter to client 103 and therefore the meter never gives permission to the tributary station to UI responses.

The HDLC based link layer is configured by means of the interface class “IEC HDLC setup” (class_id: 23, version: 1). For each physical interface one instance of this class is provided. The logical name of the instance identifies the physical interface.

1.6 | Physical Layer
Following physical interfaces support CII in a Vader device

- IEC Optical Port
- Wired M-Bus Port
- Wireless M-Bus Port (Future support :TBD)

1.7 | Scheduling
All push schedulers are synchronized to the clock; this means that all of them should be triggered at the same time. In this case they have to be scheduled to happen in order of their object definition according to the interface specification (push object 1-4).
In case that executing a scheduled push will still transmit data when another scheduled push should happen, e.g. the system is configured to have a four hour push and a five second push and the four hour push will take seven seconds to execute, the second push will not be scheduled to happen at the end. It will not be executed and re-scheduled according to its repeat setup.



1.7.1 | IEC Optical Port

On the optical port the meter is not controlling all access to the physical communication channel. It is possible, that the application running on the CII receiver can open an association to read data from the meter. One example for this would be if it needs to transfer the profile history to an attached In-home display.

1.7.1.1 | Ad-Hoc Read

On the optical port it is possible, that the external application opens an association ad hoc. In this case the communication channel is occupied and a CII Push will not be executed. The CII Push will follow the retry rules defined in its push setup.

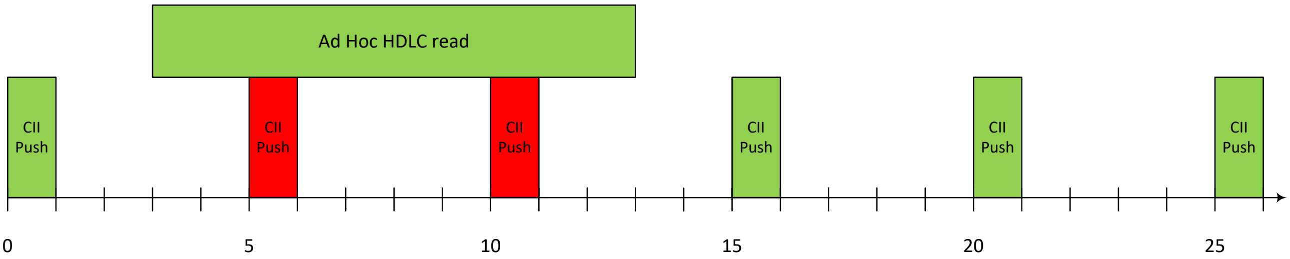


Figure 8 Ad Hoc HDLC read blocking CII Push

In case a CII Push is already underway the meter can ignore an ad hoc read, it is the client applications task to re-schedule the ad hoc read to a later time.

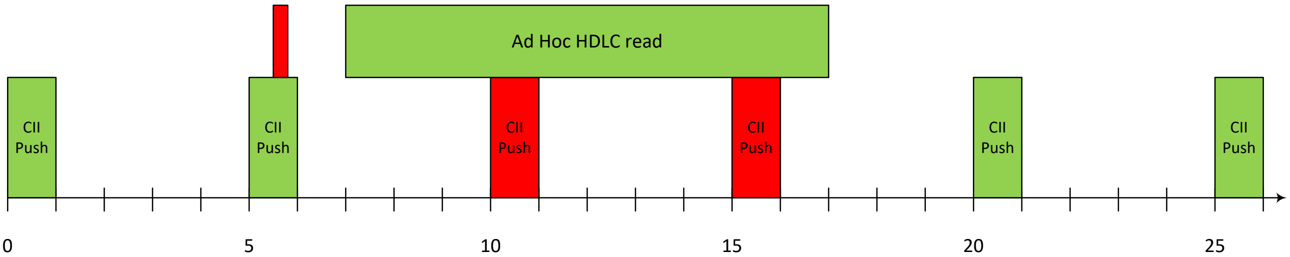


Figure 9 Ad Hoc HDLC read blocked by CII Push

Transition to/from ad hoc read and CII push should be detected and switched automatically.
If Server_enabled attribute of Security definition – CII server is set to False, No consumer information would be pushed. Server_enable attribute when set to FALSE, switch off the CII feature.

1.7.2 | Wired M-Bus

In case of wired M-Bus the meter is able to schedule the access to the physical communication channel as there are no other devices able to initiate a communication. Therefore it is able to schedule the different parties that want access. In general it is defined, that the readout of wired M-Bus devices should have priority over a CII Push. In case a push has already started when a scheduled read is supposed to happen it is acceptable though to finish the push and start the read out immediately afterwards. With an assumed accuracy of 1% the jitter in reading a one-hour value from a multi energy meter can be up to 36s. This means that, as long as the CII Push will not delay the multi energy readout by more than 36s we are still within the needed accuracy class.

1.7.2.1 | Reading of multi energy meters without clock

dMap will schedule the read of a multi energy meter that does not feature a clock short before the capture time of the multi energy profile so that we have a valid value when we come to capture the profile. As we link the CII push to the clock the chance that a longer CII Push (e.g., 24 hour profile) will happen at the same time is minimal. This means that worst case the scheduled multi energy read will be delayed by the time it takes to push the object list defined for the x seconds push which should be limited to a very few values and therefore not take too long.

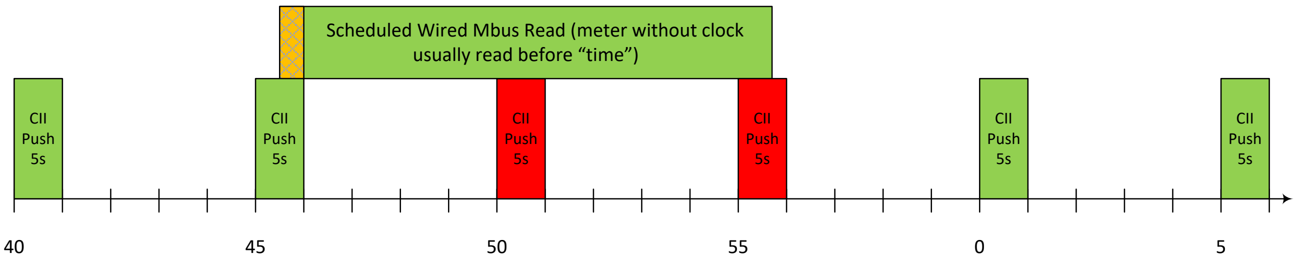


Figure 10

1.7.2.2 | Reading of multi energy meters with clock

dMap will schedule the read of a multi energy meter that features a clock short after the capture time of the profile. In this case we will read the equivalent of the last average value and apply this value to the already captured profile. In this case the delay of the read is not essential as the multi energy meter has captured its value together with a timestamp.

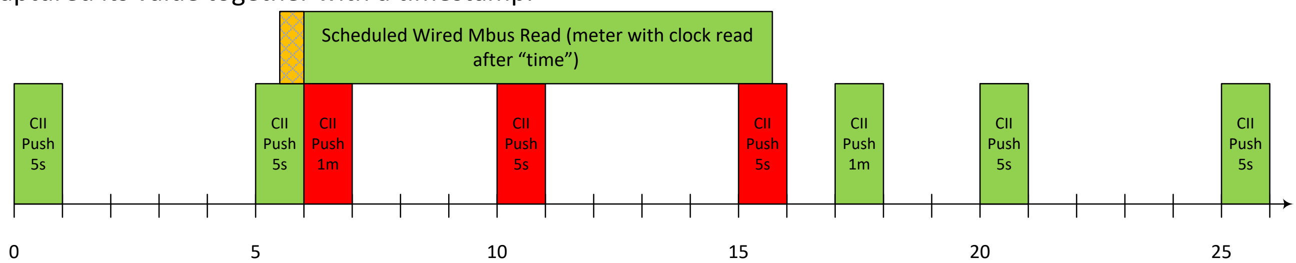


Figure 11

1.8 | Time Handling

The push schedulers are synchronized to the clock. This means that a one hour push will always happen at the full hour (i.e. at xx:00:00). This means that the push schedules are also affected by time jumps.

1.8.1 | Time setting

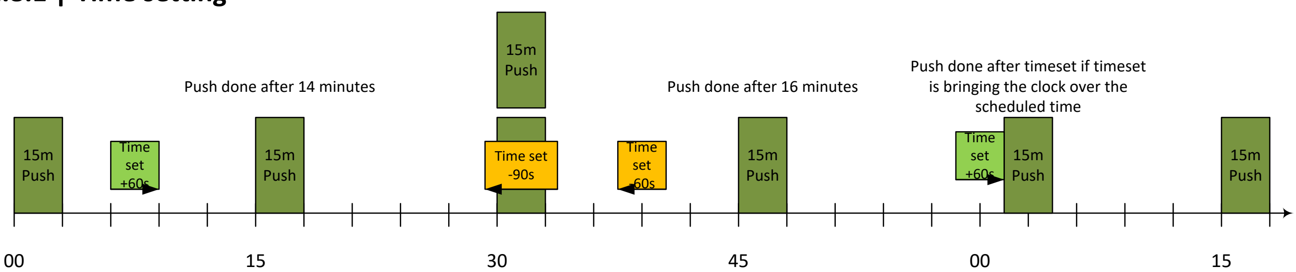


Figure 12 Time Handling

1.8.1.1 | Time setting forward

A forward time set will not affect the actual time of the next scheduled push. As shown Figure 12 Time Handling a 15 minute push schedule will happen 14 minutes after the last scheduled push in case of a 60 second forward time shift. In case the time shift is across a scheduled push, it will be executed after the time shift (see *Figure 12 Time Handling*).

1.8.1.2 | Time setting backwards

A backward time set will not affect the actual time of the next scheduled push. As shown *Figure 12 Time Handling* a 15 minute push schedule will happen 16 minutes after the last scheduled push in case of a 60 second backward time shift. In case the time shift is across a scheduled push, the push will be executed again at the appropriate time (see *Figure 12 Time Handling*).

1.8.1.3 | Time setting over multiple scheduled pushes

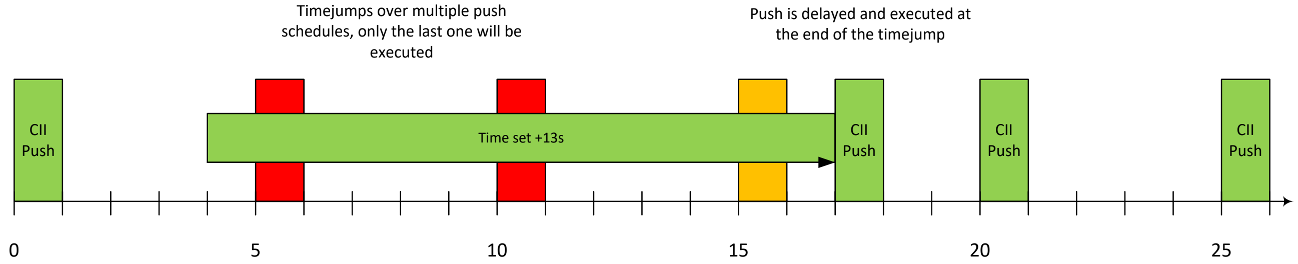


Figure 13 Time set forwards over multiple scheduled pushes

In case of a time set forwards over multiple scheduled pushes, only the last one will be executed. The execution of the last scheduled push will follow the same rules as defined in 1.8.1.1 | Time setting forward.

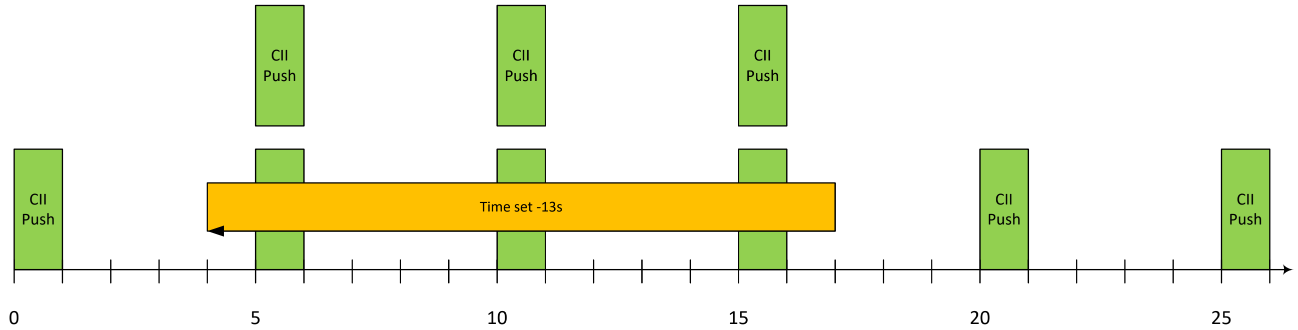


Figure 14 Time set backwards over multiple scheduled pushes

In case of a time set backwards over multiple scheduled pushes, all scheduled pushes will be executed again according to the same ruled defined in 1.8.1.2 | Time setting backwards.

1.9 | Wired Mbus as a power source

The wired Mbus provides power to the connected devices. The Mbus master in the E450 s4 and E35C 2G/3G V4.1 module is able of providing the power of 14 Mbus loads (1 Mbus load = 1.5 mA) = 20 mA at 24VDC.

The max. permanent power a connected device is allowed to consume is:

7 Mbus loads = 10 mA at 24VDC = 240 mW
The limit of 50 % of the total power has been defined to allow powering other connected wired Mbus devices.

Landis+Gyr
manage energy better

2 | Appendix:

2.1 | CIP – HAN Interface – Used Cosem Elements

Home area network (HAN) interface or Consumer Information Interface (CII) uses Consumer Information Push (CIP) to push meter data to end user. It has its own security context (security setup, frame counters, security keys). A dedicated pre-established client (Security definition 7 – Level H) with SAPID 103 is used to push data on local ports (optical and wired MBUS). Following DLMS objects are used to realize CII.

Object name	**Object-ID**	**OBIS**
Security Definition - CII Server	[30035-0-0-0]	0-0:199.35.0.255
Security setup - Consumer Information	[64-0-0-0]	0-0:43.0.1.255
IEC HDLC setup - Consumer Information Interface 1	[23-1-0-26009]	0-1:22.0.0.25
IEC HDLC setup - Consumer Information Interface 2	[23-1-0-26009]	0-2:22.0.0.255
Push action scheduler - Consumer Information	[22-0-0-26033]	0-4:15.0.4.255
Push action scheduler - Consumer Information 1 (consumption)	[22-0-0-26033]	0-5:15.0.4.255
Push action scheduler - Consumer Information 2 (TOU)	[22-0-0-26033]	0-6:15.0.4.255
Push action scheduler - Consumer Information 3 (parameterisation)	[22-0-0-26033]	0-7:15.0.4.255
Push action scheduler - Consumer Information 4 (Multiutility)	[22-0-0-26033]	0-8:15.0.4.255
Push setup - Consumer Information	[40-0-0-0]	0-6:25.9.0.255
Push setup - Consumer Information1 (consumption)	[40-0-0-0]	0-8:25.9.0.255
Push setup - Consumer Information 2 (TOU)	[40-0-0-0]	0-9:25.9.0.255
Push setup - Consumer Information 3 (parameterisation)	[40-0-0-0]	0-10:25.9.0.255
Push setup - Consumer Information 4 (Multiutility)	[40-0-0-0]	0-11:25.9.0.255