MID
SOFTWARE

SOFTWARE REQUIREMENTS AND VALIDATION GUIDE

Version 0.060
30 September 2004

Table of Contents

1 Introduction

This document provides guidance to all those concerned with the application of the Measuring Instruments Directive (MID), especially for software-equipped measuring instruments. It addresses both, manufacturers of measuring instruments and notified bodies which are responsible for conformity assessment of measuring instruments.

The Guide is purely advisory and does not itself impose any restrictions or additional technical requirements beyond those contained in the MID. Alternative approaches may be acceptable, but the guidance provided in this document represents the considered view of WELMEC as to the best practice to be followed.

The guide has been developed in the framework of the European network “MID-Software”, which was supported by the EU commission under the contract number G7RT-CT-2001-05064. Although the guide is oriented on instruments included in the regulations of the MID, the results are of a general nature and may be applied beyond.

Chapter 15 gives an overview of relation of MID-Software requirements defined in this guide to the requirements of MID.

2 Terminology

The terminology explained in this section describes the vocabulary as used in this guide. References to a standard or to any other source are given, if the definition is completely or in essential parts taken from it.

Acceptable solution: A design or a principle of a software module or hardware unit, or of a feature that is considered to comply with a particular requirement. An acceptable solution provides an example of how a particular requirement may be met. It does not prejudice any other solution that also meet the requirement.

Audit trail: A software counter (eg. “event counter”) and/or information record (eg. “event logger”) of the changes to legally relevant software or parameter.

Authentication: Verification of the declared or alleged identity of a user, process, or device.

Basic configuration: Design of the *measuring instrument* with respect to the basic architecture. There are two different basic configurations*: built-for-purpose measuring instruments* and *measuring instruments using a universal computer.* The terms are accordingly applicable to *sub-assemblies*.

Built-for-purpose measuring instrument (type P): A *measuring instrument* designed and built specially for the task in-hand. Accordingly the embedded software is assumed to be designed for the specific task. It is likely to contain many of the components also used in PCs, e.g. motherboard, memory card, etc.

Closed network: A network of a fixed number of participants with a known identity, functionality and location (see also *Open network*).

Communication interface: An electronic, optical, radio or other technical interface that enables information to be automatically passed between components of *measuring instruments or sub-assemblies*.

Device-specific parameter: *Legally relevant parameter* with a value that depends on the individual instrument. Device-specific parameters comprise calibration parameters (e.g. span adjustment or other adjustments or corrections) and configuration parameters (e.g. maximum value, minimum value, units of measurement, etc). They are adjustable or selectable only in a special operational mode of the instrument. Device-specific parameters may be classified as those that should be secured (unalterable) and those that may be accessed (settable parameters) by an authorised person, e.g. instrument owner or product vendor.

Integrated storage: non-removable storage that is part of the measuring instrument, e.g. ram, eeprom, fixed disk.

Integrity of data and software: Assurance that the data and software have not been subjected to any unauthorised changes while in use, transfer or storage.

IT configuration: Design of the *measuring instrument* with respect to IT functions and features that are – as regards the requirements – independent from the measurement function. There are four IT configurations considered in this guide: *long-term storage of measurement data*, *transmission of measurement data*, *software download* and *software separation* (see also *Basic configuration*). The terms are accordingly applicable to *sub-assemblies*.

Legally relevant parameter: Parameter of a *measuring instrument* or *a sub-assembly* subject to legal control. The following types of legally relevant parameters can be distinguished: *type-specific parameters and device-specific parameters.*

Legally relevant software: Programs, Data and *type-specific parameters* that belong to the *measuring instrument* or *sub-assembly*, and define or fulfil functions, which are subject to legal control.

Long-term storage of measurement data: Storage used for keeping measurement data ready after completion of the measurement for later legally relevant purposes (eg. the conclusion of a commercial transaction).

Measuring instrument: Any device or system with a measurement function. The adjective "measuring" is omitted if confusions can be excluded. [MID, Article 4]

Measuring instruments using a universal Computer (type U): *Measuring Instrument* that uses a general-purpose computer, usually a PC-based system, for performing legally relevant functions.

Open network: A network of arbitrary participants (devices with arbitrary functions). The number, identity and location of a participant can be dynamic and unknown to the other participants (see also *Closed network*).

Risk class: Class of *measurement instrument* types with comparable risk assessments.

Software download: The process of automatically transferring software to a target *measuring instrument* or hardware-unit using any technical means from a local or distant source (eg. exchangeable storage media, portable computer, remote computer) via arbitrary connections (eg. direct links, networks).

Software identification: A sequence of readable characters of software, and that is inextricably linked to the software (eg. version number, checksum).

Software separation: The unambiguous separation of software into *legally relevant software* and non-legally relevant software. If no software separation exists, the whole software is to consider as legally relevant.

Sub-assembly: A hardware device (hardware unit) that functions independently and makes up a *measuring instrument* together with other sub-assemblies (or a measuring instrument) with which it is compatible [MID, Article 4].

Transmission of measurement data: Transmission of measurement data via communication networks or other means to a distant device where they are further processed and/or used for legally regulated purposes.

Type-specific parameter: *Legally relevant parameter* with a value that depends on the type of instrument only. Type-specific parameters are part of the *legally relevant software*. They are fixed at type approval of the instrument.

User interface: An interface that enables information to be passed between a human user and the measuring instrument or its hardware or software components, as, e.g. switch, keyboard, mouse, display, monitor, printer, touchscreen.

Validation: Confirmation by examination and provision of objective evidence (i.e. information that can be proved true, based on facts obtained from observations, measurement, test, etc.) that the particular requirements for the intended use are fulfilled. In the present case the related requirements are those of the MID.

The following definitions are rather specific. They are only used in some extensions and for risk classes D or higher.

Hash algorithm: Algorithm that compresses the contents of a data block to a number of defined length (hash code), so that the change of any bit of the data block leads in practice to another hash code. Hash algorithms are selected such that there is theoretically a very low probability of two different data blocks having the same hash code.

Signature algorithm: A cryptographic algorithm that encrypts (encodes) plaintext to ciphertext (scrambled or secret text) using a *signature key,* and that allows decoding of the ciphertext if the corresponding *decryption signature key* is available.

Signature key: Any number or sequence of characters used encode and decode information. There are two different classes of signature keys: symmetric key systems and asymmetric key systems. Symmetric key means the sender and receiver of an information use the same key. the key system is called asymmetric if the keys for sender and receiver are different, but compatible. Usually the key of the sender is know to the sender and the key of the receiver is public in defined environment.

Public Key System (PKS): A pair of two different *signature keys*, one called the secret key and the other the public key. To verify *integrity* and *authenticit*y of information, the hash value of the information generated by a *hash algorithm* is encrypted with the secret key of the sender to create the signature, which is decrypted later by the receiver using the sender's public key

PKI Infrastructure: Organisation to guarantee the trustworthiness of a *public key system*. This includes granting and distributing digital certificates to all members that take part in the information exchange.

Certification of keys: The process of binding a public key value to an individual, organisation or other entity.

Electronic signature: A short code (the signature) that is unambiguously assigned to a text, data block or binary software file to prove the *integrity* and *authenticity* of data stored or transmitted. The signature is created using a *signature algorithm* and a secret *signature key*. Usually the generation of an electronic signature is composed of two steps: (1) first a *hash algorithm* compresses the contents of the information to be signed to a short value, and (2) then a signature algorithm combines this number with the secret key to generate the signature.

Trust Centre: An association that trustworthily generates, keeps and issues information about the authenticity of public keys of persons or other entities, e.g. measuring instruments.

3 How to use this guide

This section describes the organisation of the guide and explains how to use it.

3.1 Overall structure of the guide

The guide is organised as a structured set of requirement blocks. The overall structure of the guide follows the classification of measuring instruments into basic configurations and the classification of so-called IT configurations. The set of requirements is complemented by instrument-specific requirements.

Consequently, there are three types of requirement sets:
1. requirements for two basic configurations of measuring instruments (called type P and U),
2. requirements for four IT configurations (called extensions L, T, S and D)
3. instrument-specific requirements (called extensions I.1, I.2,…).

The first type of requirements is applicable to all instruments. The second type of requirements concerns the following IT functions: long-term storage of measurement data (L), transmission of measurement data (T), software download (D) and software separation (S). Each set of these requirements is only applicable if the corresponding function exists. The last type is a collection of further, instrument-specific requirements. The numbering follows the numbering of instrument-specific annexes in the MID. The set of requirement blocks that may be applied to a given measuring instrument is schematically shown in Figure 3-1.

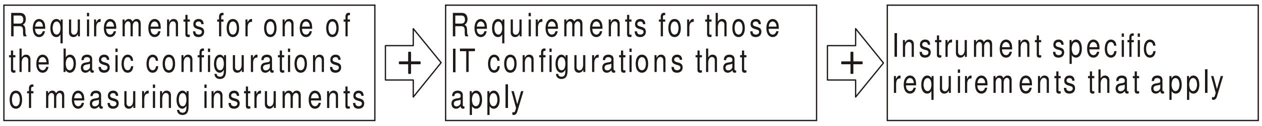
Requirements for one of the basic configurations of measuring instruments
+
Requirements for those IT configurations that apply
+
Instrument specific requirements that apply

Figure 3-1: **Type of requirement sets that should be applied to an instrument**

The schemes in the following figure 3-2 show what sets of requirements exist.

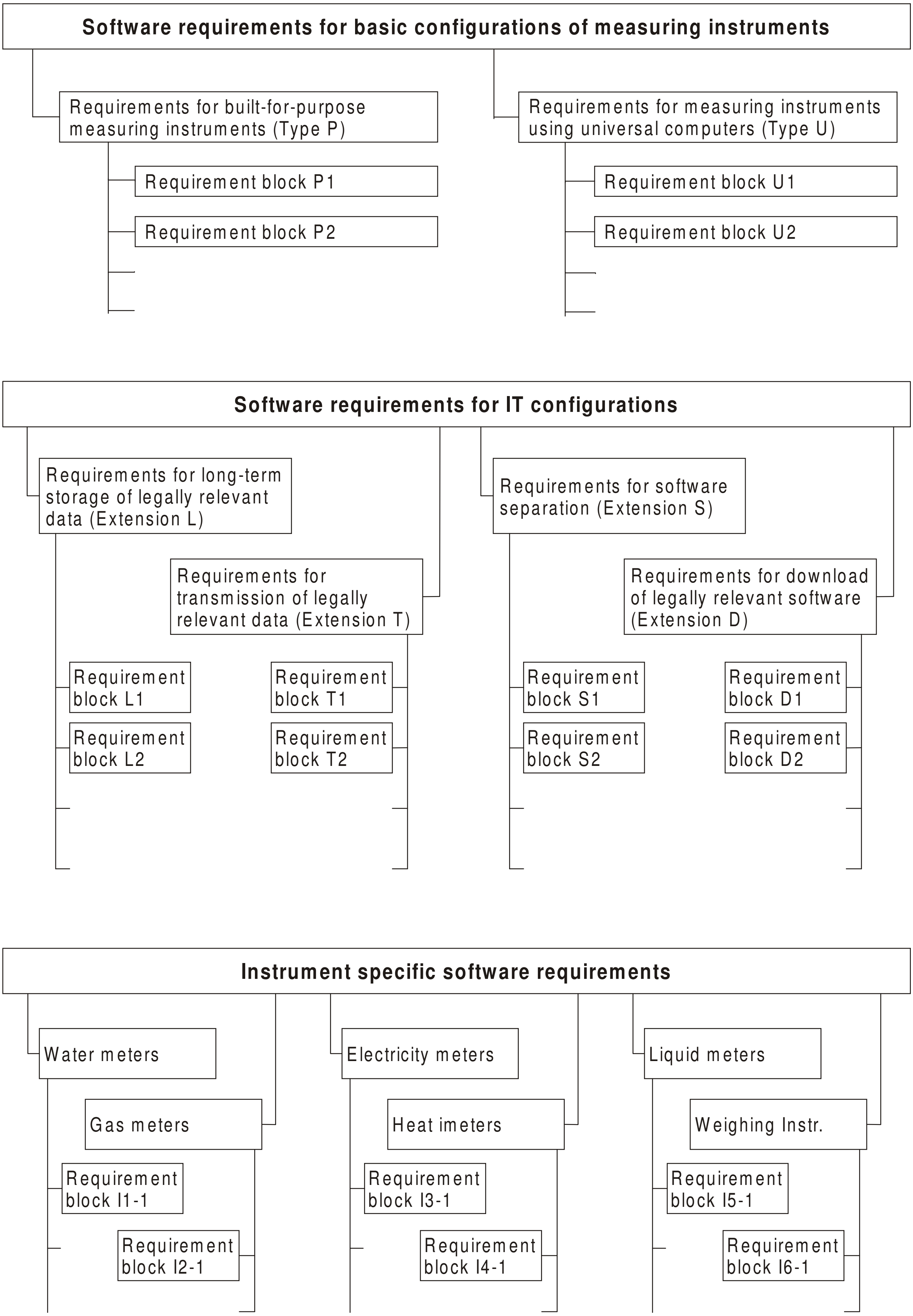
Software requirements for basic configurations of measuring instruments
Requirements for built-for-purpose measuring instruments (Type P)
Requirement block P1
Requirement block P2
Requirements for measuring instruments using universal computers (Type U)
Requirement block U1
Requirement block U2
Software requirements for IT configurations
Requirements for long-term storage of legally relevant data (Extension L)
Requirements for transmission of legally relevant data (Extension T)
Requirements for software separation (Extension S)
Requirements for download of legally relevant software (Extension D)
Requirement block L1
Requirement block L2
Requirement block T1
Requirement block T2
Requirement block S1
Requirement block S2
Requirement block D1
Requirement block D2
Instrument specific software requirements
Water meters
Gas meters
Requirement block I1-1
Requirement block I2-1
Electricity meters
Heat imeters
Requirement block I3-1
Requirement block I4-1
Liquid meters
Weighing Instr.
Requirement block I5-1
Requirement block I6-1

Figure 3-2: Overview of requirement sets

In addition to the structure described, the requirements of this guide are differentiated according to risk classes. Six risk classes, numbered from A to F with increasing risk assumptions, are introduced. The lowest risk class A and the highest risk class F are not used for the present. They are placeholders for the eventual case, that they will become necessary in future. The remaining risk classes B to E cover all of the instrument classes falling under the regulation of MID. Moreover, they provide a sufficient window of opportunity for the case of changing risk evaluations. The classes are defined in chapter 11 of this guide, which is only of an informative character.

Each measuring instrument must be assigned to a risk class because the particular software requirements to be applied are governed by the risk class the instrument belongs to.

3.2 How to select the appropriate parts of the guide

This comprehensive software guide is applicable to a large variety of instruments. The guide is modular in form. The appropriate requirement sets can be easily selected by observing the following procedure.

Step 1: Selection of the basic configuration (P or U)

Only one of the two requirement sets for basic configurations needs to be applied. Decide which basic configuration the instrument conforms to: a built-for-purpose instrument with imbedded software (type P) or an instrument using a universal computer (type U), see Figure 3-2~~.~~ (see chapter 4.1). If not the whole instrument but only a component of the instrument is the matter of concern, then decide accordingly for the component. Apply the complete set of requirements that belongs to the respective basic configuration.

Step 2: Selection of applicable IT configurations (extensions L, T, S and D)

The IT configurations comprise: long term storage of legally relevant data (L), transmission of legally relevant data (T), software separation (S) and download of legally relevant software (D). The corresponding requirement sets, called modular extensions, are independent of each other. The sets selected depend only on the IT configuration. If an extension set is selected, then it must be applied in full. Decide which, if any, of the modular extensions are applicable and apply them accordingly (Figure 3-2).

Step 3: Selection of instrument specific requirements (extension I)

Select - using the respective instrument specific extension I.x - which, if any, instrument specific requirements are applicable, and apply them accordingly (Figure 3-2).

Step 4: Selection of the applicable risk class (extension I)

Select the risk class as defined in the respective instrument specific extension I.x, sub-chapter I.x.6. There, the risk class may be defined uniformly for a class of measuring instruments or further differentiated for categories, fields of application, etc.
Once the applicable risk class has been selected, only the respective requirements and validation guidance need to be considered.

3.3 How to work with a requirement block

Each requirement block contains a well-defined requirement. It consists of a defining text, explanatory specifying notes, the documentation to be provided, the validation guidance and examples of acceptable solutions (if available). The content within a requirement block may be subdivided according to risk classes. This leads to the schematic presentation of a requirement block shown in Figure 3-3.

<table>
<tr><td colspan="3"><b>Title of the requirement</b></td></tr>
<tr><td colspan="3"><b>Main statement of the requirement</b> (eventually differentiated between risk classes)</td></tr>
<tr><td colspan="3"><b>Specifying notes</b> (scope of application, additional explanations, exceptional cases, etc.)</td></tr>
<tr><td colspan="3"><b>Documentation to be provided</b> (eventually differentiated between risk classes)</td></tr>
<tr><td><b>Validation guidance</b> for one risk class</td><td><b>Validation guidance</b> for another risk class</td><td><b>...</b></td></tr>
<tr><td><b>Acceptable solution</b> for one risk class</td><td><b>Acceptable solution</b> for another risk class</td><td>...</td></tr>
</table>

Figure 3-3: Structure of a requirement block

The requirement block represents the technical content of the requirement including the validation guidance. It addresses both~~,~~ the manufacturer and the notified body in two directions: (1) to consider the requirement as a minimal condition, and (2) not to put demands beyond this requirement.

Notes for the manufacturer:
- Observe the main statement and the additional specifying notes.
- Provide documentation as required.
- Acceptable solutions are examples that comply with the requirement. There is no obligation to follow them.
- The validation guidance has an informative character.

Notes for notified bodies:
- Observe the main statement and the additional specifying notes.
- Follow the validation guidance.
- Confirm the completeness of the documentation provided.

~~-If an acceptable solution implemented, then it can be basically accepted without further examination.~~

3.4 How to work with the checklists

Checklists are a means of ensuring that all the requirements within a chapter have been covered by the manufacturer or examiner. They are part of the pattern test report. Be aware, the checklists are only of a summarising nature, and they do not distinguish between risk classes. Checklists do not replace the requirement definitions. Refer to the requirement blocks for complete descriptions.

Procedure:
- Gather together the checklists, which are necessary according to the selection described in steps 1, 2 and 3 in section 3.2.
- Go through the checklists and prove whether all requirements have been met.
- Fill in the checklists as required.

4 Basic Requirements for Embedded Software in a Built-for-purpose Measuring Instrument (Type P)

The set of requirements of this chapter are valid for a built-for-purpose instrument or for an instrument's component that is of the built-for-purpose type. The validity for sub-assemblies is included even if it is not explicitly mentioned in the text. If the measuring instrument uses a universal computer (general purpose PC), the set of requirements in the next chapter must be referred to (Type U instrument). The requirements of the type U instrument must also be used if the subsequent technical description of built-for-purpose instruments is not matched.

4.1 Technical Description

A type P instrument is a measuring instrument with an embedded IT system (in general it is a microprocessor or microcontroller based system). It is characterised by the following features:

- The entire application software has been constructed for the measuring purpose. This includes both functions subject to legal control and other functions.
- The software is designed and treated as a whole, unless software separation according to Extension S has been observed.
- The user interface is dedicated to the measuring purpose, i.e. it is normally in an operating mode subject to legal control. Switching to an operating mode not subject to legal control is possible.
- There is no operating system having a user shell that is accessible to the user (to load programs, send commands to OS ...).
- The software and its environment are invariable and there are no means for programming or changing the legally relevant software. Software download is only allowed if extension D is observed.
- Interfaces for transmission of measurement data via open or closed communication networks are allowed (Extension T to be observed).
- The storage of measurement data either on an integrated storage, on a remote or on removable storage is allowed (Extension L to be observed).

4.2 Specific Requirements for Type P

Risk Classes B to E
P1: Documentation *In addition to the specific documentation required in each of the following requirements, the documentation shall basically include:* a. *A description of the legally relevant software.* b. *A description of the accuracy of the measuring algorithms (e.g. price calculation and rounding algorithms).* c. *A description of the user interface, menus and dialogues.* d. *The unambiguous software identification.* e. ~~*A description of data sets stored or transmitted.*~~ f. *An overview of the system hardware, e.g. topology block diagram, type of computer(s), type of network, etc, if not described in the operating manual.* g. *The operating manual.*

Risk Class B	Risk Class C	Risk Class D
P2: Software identification *The legally relevant software shall be clearly identified. An identification of the software shall be inextricably linked to the software itself. It shall be presented on command or during operation.*		
Specifying Notes: 1. Changes to legally relevant functions and characteristics of the software require a new unique software identification. The NB must be informed of the changes.	**Specifying Notes:** 1. In addition~~al~~ to 1B: Each change to legally relevant software defined as fixed at type approval require a new software identification.	
~~2.The software identification shall have a structure that clearly identifies versions modules/functions that require type approval and versions those that do not.~~ 2. The software identification shall have a structure that clearly identifies versions that require type approval and those that do not. 3. If functions of the software can be switched by type-specific parameters, each function or variant may be identified separately or, alternatively, the complete package may be identified as a whole.		
Required Documentation: The documentation shall list the software identifications and describe how the software identification is created, how it is inextricably linked to the software itself, how it may be accessed for viewing and how it is structured in order to differentiate between version changes with and without requiring a type approval**.**		**Required Documentation** (in addition to the documentation required for risk classes B and C)**:** The documentation shall show the measures taken to protect the software identification from falsification.
Validation Guidance: *Checks based on documentation:* • Examine description of the generation and visualisation of the software identification • Check whether all programs performing legally relevant functions are clearly identified and described so that it is clear to both Notified Body and manufacturer which software functions are covered by the software identification and which are not. • Check whether a nominal value of the identification (version number or functional checksum) is supplied by the manufacturer. This must be quoted in the test certificate. *Functional Checks:* • The software identification can be visualised as described in the documentation. • The presented identification is correct. The documentation (plus the executable code if necessary) of the pattern is kept at the NB.		**Validation Guidance** (in addition to the guidance for risk classes B and C)**:** *Checks based on documentation:* • Check whether the measures taken to protect from falsification are appropriate.
Acceptable Solutions: • The identification of legally relevant software comprises two parts. ~~One p~~Part (A) hast to be changed, if changes to the software require a new approval. Part (B) indicates only minor changes to the software e.g. bug fixes, which need no new approval. ~~is reserved by the manufacturer for the indication and version number, the second part (B) is identifying the legally relevant software.~~ • The identification is generated and displayed on command.		
• Part (~~B~~A) of the identification consists of a version number or the number of the TAC.	• Part (~~B~~A) of the identification consists of an automatically generated checksum over the legally relevant software that has been declared fixed at type approval. For other legally relevant software, part (~~B~~A) is a version number or the number of the TAC. • An acceptable solution for performing the checksum is the CRC-16 algorithm.	

Additions for Risk Class E
Required Documentation (in addition to the documentation required for risk classes B and C)**:** Source code that contains the generation of the identification.

Validation Guidance (in addition to the guidance for risk classes B and C)**:**

Checks based on the source code:

- Check whether all relevant software parts are covered by the algorithm for generating the identification.
- Check the correct implementation of the algorithm.

Risk Class B	Risk Class C	Risk Class D
P3: Influence via user interface *Commands entered via the user interface shall not inadmissibly influence the legally relevant software and measurement data.*		
Specifying Notes: 1. Commands may be one single or a sequence of switch or key actuations carried out manually. 2. This implies that there is an unambiguous assignment of each command to an initiated function or data change. 3. This implies that switch or key actuations that are not declared and documented as commands have no effect on the instrument's functions and measurement data.		
Required Documentation: If the instrument has the ability to receive commands, the documentation shall include: • A complete list of all commands (eg menu items) together with a declaration of completeness. • A brief description of their meaning and their effect on the functions and data of the measuring instrument.		**Required Documentation** (in addition to the documentation required for risk classes B and C)**:** • The documentation shall show the measures taken to validate the completeness of the documentation of commands. • The documentation shall contain a protocol that shows the tests of all commands.
Validation Guidance: *Checks based on documentation:* • Judge whether all documented commands are admissible, i.e. whether they have an allowed impact on the measuring functions (and relevant data) or none at all. • Check whether the manufacturer has supplied an explicit declaration of completeness of the command documentation. *Functional Checks:* • Carry out practical tests (spot checks) with both documented and undocumented commands. Test all menu items if any.		**Validation Guidance** (in addition to the guidance for risk classes B and C)**:** *Checks based on documentation:* • Check whether the measures taken and test protocols are appropriate for the high protection level.
Acceptable Solutions: There is a software module that receives and interprets commands from the user interface. This module belongs to the legally relevant software. It only forwards allowed commands to the other legally relevant software modules. All unknown or not allowed sequences of switch or key actuations are rejected and have no impact on the legally relevant software or measurement data.		

Additions for Risk Class E
Required Documentation (in addition to the documentation required for risk classes B and C)**:** Source code of the instrument.
Validation Guidance (in addition to the guidance for risk classes B and C)**:** *Checks based on the source code:* • Check the software design whether data flow concerning commands is unambiguously defined in the legally relevant software and can be verified. • Search inadmissible data flow from the user interface to domains to be protected. • Check with tools or manually that commands are decoded correctly and no undocumented commands exist.

<table>
<tr><th>Risk Class B</th><th>Risk Class C</th><th>Risk Class D</th></tr>
<tr><td colspan="3"><b>P4: Influence via communication interface</b>
<i>Commands inputted via ~~non-sealed~~ communication interfaces of the instrument ~~-~~shall not inadmissibly influence the legally relevant software and measurement data.</i></td></tr>
<tr><td colspan="3"><b>Specifying Notes:</b>
1. This implies that there is an unambiguous assignment of each command to an initiated function or data change.
2. This implies that signals or codes that are not declared and documented as commands have no effect on the instrument's functions and data.
3. Commands may be a sequence of electrical (optical, electromagnetic, etc) signals on input channels or codes in data transmission protocols.
4. The restrictions of this requirement are suspended when a software download according to Extension D is carried out.
5. This requirement applies only on interfaces which are not sealed.</td></tr>
<tr><td colspan="2"><b>Required Documentation:</b>
If the instrument has an interface the documentation shall include:
• A complete list of all commands together with a declaration of completeness.
• A brief description of their meaning and their effect on the functions and data of the measuring instrument.</td><td><b>Required Documentation</b> (in addition to the documentation required for risk classes B and C)<b>:</b>
• The documentation shall show the measures taken to validate the completeness of the documentation of commands.
• The documentation shall contain a protocol that shows the tests of the commands or alternatively any other appropriate measure to prove the correctness.</td></tr>
<tr><td colspan="2"><b>Validation Guidance:</b>
<i>Checks based on documentation:</i>
• Judge whether all documented commands are admissible, i.e. whether they have an allowed impact on the measuring functions (and relevant data) or none at all.
• Check whether the manufacturer has given an explicit declaration of completeness of the command documentation.
<i>Functional checks:</i>
• Carry out practical tests (spot checks), using peripheral equipment, if available
<i>Note:</i> If it is not possible to exclude inadmissible effects on the measurement functions (or relevant data) via the interface and the software cannot be amended accordingly, then the test certificate must indicate that the interface is non-protective and describe the securing/sealing means required. This also applies to interfaces that are not described in the documentation.</td><td><b>Validation Guidance</b> (in addition to the guidance for risk classes B and C)<b>:</b>
<i>Checks based on documentation:</i>
• Check whether the measures taken and test protocols are appropriate for the high protection level.</td></tr>
<tr><td colspan="3"><b>Acceptable Solutions:</b>
There is a software module that receives and interprets data from the interface. This module is part of the legally relevant software. It only forwards allowed commands to the other legally relevant software modules. All unknown or not allowed signal or code sequences are rejected and have no impact on the legally relevant software or measurement data.</td></tr>
</table>

<table>
<tr><th>Additions for Risk Class E</th></tr>
<tr><td><b>Required Documentation</b> (in addition to the documentation required for risk classes B and C)<b>:</b>
Source code of the instrument.</td></tr>
<tr><td><b>Validation Guidance</b> (in addition to the guidance for risk classes B and C)<b>:</b>
<i>Checks based on the source code:</i>
• Check the software design whether data flow concerning commands is unambiguously defined in the legally relevant software and can be verified.
• Search inadmissible data flow from the interface to domains to be protected.
• Check with tools or manually that commands are decoded correctly and no undocumented commands exist.</td></tr>
</table>

Risk Class B	Risk Class C	Risk Class D

P5: Protection against accidental or unintentional changes

Legally relevant software and measurement data shall be protected against accidental or unintentional changes.

Specifying Notes:

Possible reasons for accidental changes and faults are: unpredictable physical influences, effects caused by user functions and residual defects of the software even though state of the art of development techniques have been applied. This requirement includes:

a) Physical influences: Stored measurement data shall be protected against corruption or deletion when a fault occurs or, alternatively, the fault shall be detectable.
b) User functions: Confirmation shall be demanded before deleting or changing data.
c) Software defects: Appropriate measures shall be taken to protect data from unintentional changes that could occur through incorrect program design or programming errors, e.g. plausibility checks.

Required Documentation:

The documentation should show the measures that have been taken to protect the software and data against unintentional changes.

Validation Guidance:

Checks based on documentation:

- Check that a checksum of the program code and the relevant parameters is generated and verified automatically
- Check that overwriting of data cannot occur before the end of the data storage period that is foreseen and documented by the manufacturer.
- Check that a warning is issued to the user if he is about to delete measurement data files.

Functional checks:

- Check by practical spot checks that before deleting measurement data a warning is given, if deleting is possible at all.

Acceptable Solutions:

- The accidental modification of software and measurement data may be checked by calculating a checksum over the relevant parts, comparing it with the nominal value and stopping if anything has been modified.
- Measurement data are not deleted without prior authorisation, e.g. a dialogue statement or window asking for confirmation of deletion.
- For fault detection see also Extension I.

Additions for Risk Class E

Required Documentation (in addition to the documentation required for risk classes B, C and D)**:**

Source code of the instrument.

Validation Guidance (in addition to the guidance for risk classes B, C and D)**:**

Checks based on the source code:

- Check whether measures taken for detection of changes (faults) are appropriate.
- If a checksum is realised, check whether all parts of the legally relevant software are covered by it.

<table>
<tr><th>Risk Class B</th><th>Risk Class C</th><th>Risk Class D</th></tr>
<tr><td colspan="3">**P6: Protection against intentional changes**
Legally relevant software shall be secured against the inadmissible modification, loading or swapping of hardware memory.</td></tr>
<tr><td colspan="3">**Specifying Notes:**
1. Instrument without interface: Manipulation of program code could be possible by manipulating the physical memory, i.e. the memory is physically removed and substituted by one containing fraudulent software or data. To prevent this happening, either the housing of the instrument should be secured or the physical memory itself is secured against unauthorised removal.
2. Instrument with interface: The interface shall include only functions, which are subject to examination. All functions in the interface shall be subject to examination (see P4). Where the interface is to be used for software download, extension D must be complied with.
3. Data are considered to be sufficiently protected if only legally relevant software processes them. If non-legally relevant Software is intended to be changed after approval, requirements of extension S have to be followed.</td></tr>
<tr><td colspan="2">**Required Documentation:**
The documentation shall provide assurance that legally relevant software cannot be inadmissibly modified.</td><td>**Required Documentation** (in addition to the documentation required for risk classes B and C)**:**
The protection measures taken to protect from intentional changes shall be shown.</td></tr>
<tr><td colspan="2">**Validation Guidance:**
Checks based on documentation:
• Examine whether the documented means of securing against unauthorised exchange of the memory that contains the software are sufficient.
• If the memory can be programmed in-circuit (without dismounting), check whether the programming mode can be disabled electrically and the means for disabling can be secured/sealed. (For checking download facilities see extension D)
Functional checks
• Test practically the programming mode and check whether disabling works.</td><td>**Validation Guidance** (in addition to the guidance for risk classes B and C):
Checks based on documentation:
• Check whether the measures taken are appropriate with respect to the required state of the art for a high protection level.</td></tr>
<tr><td colspan="3">**Acceptable Solutions:**
The instrument is sealed and the interfaces comply with the requirements P3 and P4.</td></tr>
</table>

Additions for Risk Class E
Required Documentation (in addition to the documentation required for risk classes B and C)**:** Source code of the instrument.
Validation Guidance (in addition to the guidancefor risk classes B and C)**:** *Checks based on the source code:* • Check in the source code whether measures taken for the detection of intentional changes are appropriate.

<table>
<tr><th>Risk Class B</th><th>Risk Class C</th><th>Risk Class D</th></tr>
<tr><td colspan="3"><b>P7: Parameter protection</b>
<i>Parameters that fix legally relevant characteristics of the measuring instrument shall be secured against unauthorised modification.</i></td></tr>
<tr><td colspan="3"><b>Specifying Notes:</b>
1. Type specific parameters are identical for each specimen of the type and are in general part of the program code. Therefore requirement P6 applies to them.
2. Device specific secured parameters may be changed using an on-board keypad or switches or via interfaces, but only before they have been secured.
3. Settable device-specific parameters may be changed after securing.</td></tr>
<tr><td colspan="2"><b>Required Documentation:</b>
The documentation should describe all of the legally relevant parameters, their ranges and nominal values, where they are stored, how they may be viewed, how they are secured and when, i.e. before or after verification.</td><td><b>Required Documentation</b> (in addition to the documentation required for risk classes B and C)<b>:</b>
The protection measures taken for parameters shall be shown.</td></tr>
<tr><td colspan="2"><b>Validation Guidance:</b>
<i>Checks based on documentation:</i>
• Check that changing or adjusting secured device specific parameters is impossible after securing.
• Check whether all relevant parameters according to the lists (given in Extension I, if any) have been classified as secured.
<i>Functional checks:</i>
• Test the adjusting (configuration) mode and check whether disabling after securing works.
• Examine the classification and state of parameters (secured/settable) at the display of the instrument, if a suitable menu item is provided.</td><td><b>Validation Guidance</b> (in addition to the guidance for risk classes B and C):
<i>Checks based on documentation:</i>
• Check whether the measures taken are appropriate with respect to the required state of the art for a high protection level.</td></tr>
<tr><td colspan="3"><b>Acceptable Solutions:</b>
a) Parameters are secured by sealing the instrument or memory housing and disabling the write enable/disable input of the memory circuit by an associated jumper or switch, which is sealed.</td></tr>
<tr><td colspan="2"><i>Audit trails:</i>
b) An <b>event counter</b> registers each change of a parameter value. The current count can be displayed and can be compared with the initial value of the counter that was registered at the last official verification and is indelibly labelled on the instrument.
c) Changes of parameters are registered in an <b>event logger</b>. It is an information record stored in a non-volatile memory. Each entry is generated automatically by the legally relevant software and contains:
• the identification of the parameter (eg the name)
• the parameter value (the current or the value before)
• the time stamp of the change
The event logger cannot be deleted or be changed in another way than by the legally relevant software without destroying a seal.</td><td>÷</td></tr>
</table>

Risk Class E
Required Documentation (in addition to the documentation required for risk classes B and C)**:** Source code showing the way of securing and viewing legally relevant parameters.
Validation Guidance (in addition to the guidance for risk classes B and C)**:** *Checks based on the source code:* • Check in the source code whether measures taken for protecting parameters are appropriate (e.g. adjusting mode disabled after securing).

5 Basic Requirements for Software of Measuring Instruments using a Universal Computer (Type U)

5.1 Technical Description

The set of software requirements in this section apply to a measuring instrument based on a general-purpose computer. The technical description of the Type U measuring system is summarised in Table 5.1 below. Basically, a Type U system must be assumed if the conditions of a type P instrument (see chapter 4.1) are not fulfilled.

Description
Hardware Configuration a. A modular general-purpose computer-based system. The computer system may be stand-alone, part of a closed network, e.g. Ethernet, token-ring LAN, or part of an open network, e.g. Internet. b. Because the system is general purpose, the sensor would normally be external to the computer unit and would normally be linked to it by a closed communications link. The communication link could, however, also be open, e.g. network, whereby several sensors could be connected. c. The user interface may be switched from an operating mode, which is not under legal control, to one which is, and vice-versa. d. Storage may be local, e.g. hard disk, or remote, e.g. file server. Remote storage may be located anywhere, e.g. in the same building or even in a different country, which could be outside the EU. Thus the communications link to storage devices may be direct, which permits handshaking, or indirect, whereby there might be an intermediate storage phase not under the control of the user, e.g. dial-up on Internet. Storage may be fixed, e.g. hard disk, or removable, e.g. diskettes, CD-RW.
Software Configuration e. Any operating system may be used. In addition to the measuring instrument application, other software applications may also reside on the system at the same time. Parts of the software, e.g. measuring instrument application, are subject to legal control and may not be inadmissibly modified after approval. Parts not subject to legal control may be freely modified. f. The operating system and low level drivers, e.g. video drivers, printer drivers, disk drivers, etc., are not legally relevant unless they are specially programmed for a specific measuring task.

Table 5.1 Technical description of a Type-U measuring instrument.

5.2 Specific Software Requirements for Type U

Risk Classes B to E
U1: Documentation *In addition to the specific documentation required in each requirement below, the documentation shall basically include:* a. *A description of the legally relevant software functions, meaning of the data, etc.* b. *A description of the accuracy of the measuring algorithms (e.g. price calculation and rounding algorithms).* c. *A description of the user interface, menus and dialogues.* d. *A legal software identification.* e. ~~*A description of data sets of stored or transmitted data.* Why do we need a description of this data? Surely, what we really need to know is whether legally relevant data is stored on removable media or transmitted to remote storage.~~ f. *An overview of the system hardware, e.g. topology block diagram, type of computer(s), type of network, etc, if not described in the operating manual.* g. *An overview of the security aspects of the operating system, e.g. protection, user accounts, privileges, etc.* h. *The operating manual.*

<table>
<tr><th>Risk Class B</th><th>Risk Class C</th><th>Risk Class D</th></tr>
<tr><td colspan="3"><b>U2: Software identification</b>
<i>The legally relevant software shall be clearly identified. An identification of the software shall be inextricably linked to the software itself. It shall be determined and presented on command or during operation.</i></td></tr>
<tr><td><b>Specifying Notes:</b>
1. Identification excludes the operating system and low level drivers, e.g. video drivers, printer drivers, disk drivers, etc. but it does include drivers specially programmed for a specific legally relevant task.
2. Changes to metrologically relevant software require a new unique software identification. The NB must be informed of the changes.</td><td colspan="2"><b>Specifying Notes:</b>
1. Restriction of 1B: (Low level) drivers that are defined as relevant at type approval shall be identified.
2. Additional to 2B: Each change to legally relevant program code defined as fixed at type approval or changes of type-specific parameters require a new software identification.</td></tr>
<tr><td colspan="3">~~3.The software identification shall have a structure that clearly identifies versions~~ modules/functions ~~that require type approval and version~~ those ~~that do not.~~
3. The software identification shall have a structure that clearly identifies versions that require type approval and those that do not.
4. Identifications may be applied to different levels, e.g. to complete programs, modules, functions, etc.
5. If functions of the software can be switched by parameters, each function or variant may be identified separately or the complete package may be identified as a whole.</td></tr>
<tr><td colspan="2"><b>Required Documentation:</b>
The documentation shall list the software identifications and describe how the software identification is created, how it is inextricably linked to the software itself, how it may be accessed for viewing and how it is structured in order to differentiate between version changes with and without requiring a type approval.</td><td><b>Required Documentation</b> (in addition to the documentation required for risk classes B and C)<b>:</b>
The documentation shall show the measures taken to protect the software identification from falsification.</td></tr>
</table>

Validation Guidance:

Checks based on documentation:

- Examine description of the generation and visualisation of the software identification
- Check whether all legally relevant software is clearly identified and described so that it should be clear to both Notified Body and manufacturer which software functions are covered by the software identification and which are not.
- Check whether a nominal value of the identification (version number or functional checksum) is supplied by the manufacturer. This must be quoted in the test certificate.

Functional checks:

- Check whether the software identification can be visualised as described in the documentation.
- Check whether the presented identification is correct.

The documentation (plus the executable code if necessary) of the pattern is kept at the NB.

Validation Guidance (in addition to the documentation required for risk classes B and C)**:**

Checks based on documentation

- Check whether the measures taken to protect from falsification are appropriate.

Acceptable Solutions:

- The identification of legally relevant software comprises two parts. Part (A) hast to be changed, if changes to the software require a new approval. Part (B) indicates only minor changes to the software e.g. bug fixes, which need no new approval.~~The identification of legally relevant software comprises two parts. One part (A) is reserved by the manufacturer for the indication title? and version number, the second part (B) is identifying the legally relevant software.~~
- The identification part (B) is generated and displayed on command.

- Part (~~B~~A) of the identification consists of a version number or the number of the TAC. To prevent it from being changed with simple software tools, it is stored in binary format in the executable program file.

- Part (A~~B~~) of the identification consists of an automatically generated checksum over the fixed code. For other legally relevant software, part (~~B~~A) is a version number or the number of the TAC. To prevent it from being changed with simple software tools, it is stored in binary format in the executable program file.

- An acceptable solution for performing the checksum is the CRC-16.

- Acceptable algorithms for the checksum are CRC-32 or hash algorithms like SHA-1, MD5, RipeMD160 etc.

Additions for Risk Class E

Required Documentation (in addition to the documentation required for risk classes B and C)**:**
Source code that contains the generation of the identification.

Validation Guidance (in addition to the guidance for risk classes B and C)**:**

Checks based on the source code:

- Check whether all relevant software parts are covered by the algorithm for generating the identification.
- Check the correct implementation of the algorithm.

Risk Class B	Risk Class C	Risk Class D

U3: Influence via user interfaces

Commands entered via the user interface shall not inadmissibly influence legally relevant software and measurement data.

Specifying Notes:

1. This implies that there is an unambiguous assignment of each command to an initiated function or data change.
2. This implies that switch or key actuations that are not declared and documented as commands have no effect on the instrument's functions and measurement data.
3. Commands may be a single action or a sequence of actions carried out by the operator. The user shall be guided which commands are allowed.

÷

4. The user shell shall be closed ie the user shall not be able to load programs, write programs or perform commands to the operating system.

<table>
<tr><td><b>Required Documentation:</b>
The documentation shall include:
• A complete list of all commands together with a declaration of completeness.
• A brief description of their meaning and their effect on the functions and data of the measuring instrument.</td><td><b>Required Documentation</b> (in addition to the documentation required for risk classes B and C)<b>:</b>
• The documentation shall show the measures to validate the completeness of the documentation of commands.
• The documentation shall contain a protocol that shows the tests of all commands.</td></tr>
<tr><td><b>Validation Guidance:</b>
<i>Checks based on documentation:</i>
• Judge that documented commands are admissible, i.e. that they have an allowed impact on the measuring functions (and relevant data) or none at all.
• Check that manufacturer has supplied an explicit declaration of completeness of the command documentation.
<i>Functional checks:</i>
• Carry out practical tests (spot checks) with both documented and undocumented commands. Test all menu items if any.</td><td><b>Validation Guidance</b> (in addition to the guidance for risk classes B and C)<b>:</b>
<i>Checks based on documentation:</i>
• Check whether the measures taken and test protocols are appropriate for the high protection level.</td></tr>
<tr><td colspan="2"><b>Acceptable Solution:</b>
• A module in the legally relevant software filters out inadmissible commands. Only this module receives commands, and there is no circumvention of it. Any false input is blocked. The user is controlled or guided when inputting commands by a special software module. This guiding module is inextricably linked with the module that filters out the inadmissible commands.</td></tr>
<tr><td>÷</td><td>• The access to the operating system is blocked.</td></tr>
</table>

Additions for Risk Class E
Required Documentation (in addition to the documentation required for risk classes B and C)**:** Source code of the legally relevant software.
Validation Guidance (in addition to the guidance for risk classes B and C)**:** *Checks based on the source code:* • Check the software design whether data flow concerning commands is unambiguously defined in the legally relevant software and can be verified. • Search inadmissible data flow from the user interface to domains to be protected. • Check with tools or manually that commands are decoded correctly and no undocumented commands exist.

<table>
<tr><th>Risk Class B</th><th>Risk Class C</th><th>Risk Class D</th></tr>
<tr><td colspan="3"><b>U4: Influence via communication interface</b>
<i>Commands input via non-sealed communication interfaces of the device shall not inadmissibly influence the legally relevant software and measurement data.</i></td></tr>
<tr><td colspan="3"><b>Specifying Notes:</b>
1. This implies that there is an unambiguous assignment of each command to an initiated function or data change.
2. This implies that signals or codes that are not declared and documented as commands have no effect on the instrument's functions and data.
3. Commands may be a sequence of electrical (optical, electromagnetic, etc) signals on input channels or codes in data transmission protocols.
4. The restrictions of this requirement are suspended when a software download according to Extension D is carried out.</td></tr>
<tr><td colspan="2">5. Those parts of the operating system that interpret legally relevant commands shall be considered to be legally relevant software.
6. Other software parts may use the interface provided they do not disturb or falsify the reception or transmission of legally relevant commands or data.</td><td>5. All programs and program parts involved in the transmission and reception of legally relevant commands or data shall be supervised by the legally relevant software.
6. The interface that receives or transmits legally relevant commands or data shall be dedicated to</td></tr>
</table>

	that role and may be used only by legally relevant software. Standard interfaces are not excluded, however, if software protection means are implemented according to extension T. ~~Is this practical? On a U-type we cannot stop someone sending non-legally relevant software to this interface when it is in service unless it is physically sealed. Instead, we must rely on the efficiency of the filter. The restrictions of this requirement are suspended when a transmission complying with the requirements of Extension T is carried out.~~
Required Documentation: The documentation shall include: • A complete list of all commands together with a declaration of completeness. • A brief description of their meaning and their effect on the functions and data of the measuring instrument.	**Required Documentation** (in addition to the documentation required for risk classes B and C)**:** • The documentation shall show the measures taken to validate the completeness of the documentation of commands. • The documentation shall contain a protocol that shows the tests of the commands or alternatively any other appropriate measure to prove the correctness.
Validation Guidance: *Checks based on documentation:* • Judge whether all documented commands are admissible, i.e. whether they have an allowed impact on the legally relevant software (and relevant measurement data) or none at all. • Check whether the manufacturer has given an explicit declaration of completeness of the command documentation. *Functional checks:* • Carry out practical tests (spot checks), using peripheral equipment, if available	**Validation Guidance** (in addition to the guidance for risk classes B and C)**:** *Checks based on documentation:* • Check whether the measures taken and test protocols are appropriate for the high protection level.

Acceptable Solutions:
There is a software module that receives and interprets commands from the interface. This module belongs to the legally relevant software. It only forwards allowed commands to the other legally relevant software modules All unknown or not allowed commands are rejected and have no impact on the legally relevant software or measurement data.

Additions for Risk Class E
Required Documentation (in addition to the documentation required for risk classes B and C)**:** Source code of the instrument.
Validation Guidance (in addition to the guidance for risk classes B and C)**:** *Checks based on the source code:* • Check the software design whether data flow concerning commands is unambiguously defined in the legally relevant software and can be verified. • Search inadmissible data flow from the interface to domains to be protected. • Check with tools or manually that commands are decoded correctly and no undocumented commands exist.

Risk Class B	Risk Class C	Risk Class D
U5: Protection against accidental or unintentional changes *Legally relevant software and measurement data shall be protected against accidental or unintentional changes.*		

Specifying Notes:

1. Unintentional changes could occur through:
   a. Incorrect program design, e.g. incorrect loop operation, changing global variables in a function, etc.
   b. Misuse of the operating system
   c. Accidental overwriting or deletion of stored data and programs (refer also to Extension L).
   d. Incorrect assignment of measurement transaction data. Measurements and data belonging to one measurement transaction must not be mixed with those of a different transaction due to incorrect programming or storage.
   e. Physical effects (electromagnetic interference, temperature, vibration, etc).

Required Documentation:

The documentation should show the measures that have been taken to protect the software and data against unintentional changes.

Required Documentation (in addition to the documentation required for risk classes B and C)**:**

The documentation shall show the measures taken to validate the effectiveness of the protection means.

Validation Guidance:

Checks based on documentation:

- Check that a checksum of the program code and the relevant parameters is generated and checked automatically
- Check that overwriting of data cannot occur before the end of the data storage period that is foreseen and documented by the manufacturer.
- Check that a warning is issued to the user if he is about to delete measurement data files.

Functional checks:

- Check by practical spot checks that before deleting measurement data a warning is given, if deleting is possible at all.

Validation Guidance (in addition to the guidance for risk classes B and C)**:**

Checks based on documentation:

- Check whether the measures taken are appropriate for the high protection level.

Acceptable Solutions:

- Prevention from incorrect program design – this is beyond the scope of these risk classes.
- Misuse of the operating system, overwriting or deletion of stored data and programs – the manufacturer should make full use of the protection or privacy rights provided by the operating system or programming language.
- The accidental modification of programs & data files may be checked by calculating a checksum over the relevant code, comparing it with the nominal value and stopping if the code has been modified or suitably reacting, if parameters or data are concerned.
- Where the operating system allows it, it is recommended that all user rights for the deletion, moving or amendment of legally relevant software should be removed and access should be controlled via utility programs. Access control to programs and data through the use of passwords is recommended, as is the use of read-only mechanisms. The system supervisor should restore rights only when required.

Additions for Risk Class E

Required Documentation (in addition to the documentation required for risk classes B and C)**:**
Source code of the instrument.

Validation Guidance (in addition to the guidance for risk classes B and C)**:**

Checks based on the source code:

- Check whether measures taken for detection of changes (faults) are appropriate.
- If a checksum is realised, check whether all parts of the legally relevant software are covered by it.

Risk Class B	Risk Class C	Risk Class D

U6: Protection against intentional changes
Legally relevant software and measurement data shall be secured against inadmissible modification.

Specifying Notes:

1. Changes with the intention of fraud could be attempted by:
   a. Changing the program code including integrated data - if the program code is an executable format (.exe) then it is sufficiently protected for risk classes B and C.
   b. Changing the measurement data – refer to Extension L.

Specifying Notes:

1. The level of protection should be equivalent to that of electronic payments.

In general, a universal computer is

2. Exchange of the approved software shall not be possible simply by using the operating system e.g. to load and use non-approved software instead (see U3). For downloading software see Extension D. 3. Where necessary, means shall be taken to protect legally relevant software against modification by simple tools, e.g. text or window editors.	only suitable for this risk class with additional hardware for securing.
Required Documentation: The documentation should provide assurance that software and stored measurement data cannot be inadmissibly modified.	**Required Documentation** (in addition to the documentation required for risk classes B and C)**:** The protection measures taken shall be shown.
Validation Guidance: **Case 1:** Closed shell of the software subject to legal control. *Checks based on documentation:* • Software modules boot automatically. • User has no access to the operating system of the PC. • User has no access to other software than the approved one. • A written declaration is given that there are no hidden functions to circumvent the closed shell. **Case 2:** User-accessible operating system and/or software. *Checks based on documentation:* • Checksum over machine code of the software modules is generated. Legally relevant software cannot be started if code is falsified.	**Validation Guidance** (in addition to the guidance for risk classes B and C)**:** *Checks based on documentation:* • Check whether the measures taken are appropriate with respect to the required state of the art for a high protection level.
Acceptable Solutions: • Program code and data may be protected by means of checksums. The program is calculating its own checksum and compares is with a desired value that is hidden in the executable code. If the self-check fails, the program is blocked. • Any signature algorithm should have a key length of at least 2 bytes; a CRC-16 checksum with a secret initial vector (hidden in the executable code) would be satisfactory. (See also Extensions L and T). • The unauthorised manipulation of legally relevant software may be controlled by the access control or privacy protection attributes of the operating system. The administration level of these systems shall be secured by sealing or equivalent means.	**Acceptable Solutions:** • Program code may be secured by storing the legally relevant software in a dedicated plug-in-unit, which is sealed. The plug-in unit may include, for example, a read-only memory and a microcontroller.

Additions for Risk Class E
Required Documentation (in addition to the documentation required for risk classes B and C)**:** Source code of the instrument.
Validation Guidance (in addition to the guidance required for risk classes B and C)**:** *Checks based on the source code:* • Check communication with the additional securing hardware. • Check that changes of programs or data are detected and program execution stops in this case.

Risk Class B	Risk Class C	Risk Class D
U7: Parameter protection *Legally relevant parameters shall be secured against unauthorised modification.*		
Specifying Notes: 1. Type specific parameters are identical for each specimen of the type and are in general part of the program code ie part of the legally relevant software. Therefore requirement U6 applies to them. 2. Device specific parameters: • “Secured” parameters may be changed using an on-board keypad or switches or via interfaces but only *before* the action of securing. Because device specific parameters could be manipulated using simple tools *on universal computers they shall not be stored in standard storages of a universal computer*. Storing of		

these parameters is acceptable only in additional hardware.
- Settable device specific parameters may be changed after securing.

Required Documentation:

The documentation shall describe all of the legally relevant parameters, their ranges and nominal values, where they are stored, how they may be viewed, how they are secured and when, i.e. before or after verification.

Required Documentation (in addition to the documentation required for risk classes B and C)**:**

The protection measures taken for parameters shall be shown.

Validation Guidance:

Checks based on documentation:

- Check that the method for protection of the type specific parameters is appropriate.
- Check that device specific parameters are not stored on the standard storages of the universal computer but in separate hardware that can be sealed and write-disabled.

Functional checks:

- Test the adjusting (configuration) mode and check whether disabling after securing works.
- Examine the classification and state of parameters (secured/settable) at the display of the instrument, if a suitable menu item is provided.

The TAC should list those parameters that are settable and how they may be located.

Validation Guidance (in addition to the guidance for risk classes B and C)**:**

Checks based on documentation:

- Check whether the measures taken are appropriate with respect to the required state of the art for a high protection level.

Acceptable solutions:

- Device specific parameters are stored on a plugged-in storage which is sealed against removing or directly on the sensor unit. Writing of parameters is inhibited by sealing a write-enable switch in the disabled state. Audit trails are possible in combination with securing hardware (see P7).
- Settable parameters are stored on a standard storage of the universal computer.

Additions for Risk Class E

Required Documentation (in addition to the documentation required for risk classes B and C)**:**
Source code of the instrument.

Validation Guidance (in addition to the guidance for risk classes B and C)**:**

Checks based on the source code:

- Check whether measures taken for protecting parameters are appropriate.

Risk Class B	Risk Class C	Risk Class D

U8: Software authenticity and presentation of results

Means shall be employed to ensure the authenticity of the legally relevant software. The authenticity of the results that are presented shall be guaranteed.

Specifying Notes:

1. It shall not be possible to fraudulently simulate (spoof) approved legally relevant software using simple software tools.
2. Presented results can be accepted as authentic if the presentation is issued from within the legally relevant software

Specifying Notes:

1. Restriction to 1BC, 2BC: Means are required to protect against intentional misuse, including simulation, based on additional hardware.

3. Presented measurement values shall be accompanied by any information necessary to avoid confusion with other (non-legally relevant) information.
4. It shall be ensured by technical means that on the universal computer only the software approved for the legally relevant purpose can perform the legally relevant functions (eg a sensor shall only work together with the approved program).

Required Documentation: The documentation should describe how authenticity of the software is guaranteed.	**Required Documentation** (in addition to the documentation required for risk classes B and C)**:** The protection measures taken shall be shown.
Validation Guidance: *Checks based on documentation:* • The examination needs to determine that presentations are generated by legally relevant software and how spoofing by non-legally relevant programs may be prevented. • Check that the legally relevant tasks can only be performed by the approved legally relevant software. *Functional checks:* • Check through visual control if the presentation of results is easily distinguishable from other information that may also be presented. • Check according to the documentation if the presented information is complete.	**Validation Guidance** (in addition to the guidance for risk classes B and C)**:** *Checks based on documentation:* • Check whether the measures taken are appropriate with respect to the required state of the art for a high protection level.

Acceptable Solutions:

Formal means:

1. The software identification part (B) (checksum, version number or TAC number, see U2) indicated by the software is compared with the desired value in the TAC.

Technical means:

1. A measurement application window is generated by the legally relevant software. The technical measures required of the window are:
   - No access to measurement values shall be given to non-legally relevant programs until the measurement values have been indicated.
   - The window is refreshed periodically. The associated program checks that it is always visible.
   - Processing of measurement values stops whenever this window is closed or not completely visible.

   The operating manual (and TAC) should contain a copy of the window for reference purposes.

2a The sensor unit encrypts the measuring values with a key known to the approved software running on the universal computer (e.g. ist version number). Only the approved software can decrypt and use the measurement values, non-approved programs on the universal computer cannot as they don’t know the key. For key treatment see Extension T.

2b Before sending measurement values the sensor initiates a handshake sequence with the legally relevant software on the universal computer based on secret keys. Only if the program on the universal computer communicates correctly, the sensor unit sends its measurement values. For key treatment see Extension T.

3. The key used in 2a / 2b may be chosen and entered to the sensor unit and software on the universal computer without destroying a seal.	3. The key used in 2a / 2b is the hash code of the program on the universal computer. Each time the software on the universal computer is changed, the new key has to be entered into the sensor unit and sealed.

Additions for Risk Class E

Required Documentation (in addition to the documentation required for risk classes B and C)**:**
Source code of the legally relevant software.

Validation Guidance (in addition to the guidance for risk classes B and C)**:**

Checks based on the source code:

- Check that legally relevant software generates the presented measurement results.
- Check whether all measures taken are appropriate and correct to guarantee the authenticity of the software (e.g. that legally relevant tasks can only be performed by the approved legally relevant software).

Risk Classes B to E

U9: Influence of other software

The legally relevant software shall be designed in such a way that other software does not inadmissibly influence it.

Specifying Notes:

This requirement implies software separation between the legally relevant and non-legally relevant software. Extension S shall be observed. This is the standard case for universal computers.

Required Documentation: See Extension S.
Validation Guidance: See Extension S.
Acceptable Solutions: See Extension S.

6 Extension L: Long-term Storage of Measurement Data

This is an extension to the specific requirements of embedded software for built-for-purpose measuring instrument (type P requirements) and of software for measuring instruments using a universal computer (type U requirements). It describes the requirements for the storage of measurement data from when a measurement is physically completed to the point in time when all processes to be done by the *legally relevant software are* finished. It may also be applied to long-term storage of the data thereafter.

6.1 Technical description

The set of requirements of this extension only apply if long-term storage of measurement data is included. It concerns only those measurement date that are legally relevant. Three different technical configurations for long-term storage are listed in the following table. For a built-for-purpose device, the variant of an integrated storage is typical: here the storage is part of the metrologically necessary hardware and software. For instruments using a universal computer, another variant is typical: the use of resources already existing, e.g., hard disks. The third variant is the removable storage: here the storage can be removed from the device, which could be either a built-for-purpose device or a universal computer, and be taken elsewhere. When data is retrieved from removable storage for legal purposes, e.g. visualisation, ticket printing, etc, the retrieving device shall be subject to legal control.

Description of technical configuration
Integrated storage Simple instrument, built-for-purpose, no externally usable tools or means available for editing or changing data, integrated storage for measurement data or parameters, e.g. RAM, flash memory, hard disk.
Storage for universal computer Universal computer, graphical user interface, multitasking operating system, tasks subject to legal control and not subject to legal control exist in parallel, storage can be removed from the device or contents can be copied anywhere inside or outside the computer.
Removable or remote (external) storage Arbitrary basic instrument (built-for-purpose instrument or instrument using universal computer), storage can be taken from the instrument. These can be, for example, floppy disks, flash cards, or remote databases connected via network.

6.2 Specific software requirements for Long-term Storages

The requirements given in this section are to apply in addition to one set of requirements, either for the basic built-for-purpose instruments or for instruments using universal computer.

Risk Class B	Risk Class C	Risk Class D
L1 Completeness of measurement data stored *The measurement data stored must contain all relevant information necessary to reconstruct an earlier measurement.*		
Specifying Notes: 1. The stored measurement data may be needed for reference at a later date e.g. for checking invoices. All data necessary for legal and metrological reasons shall be stored together with the measurement value.		
Required Documentation: Description of all fields of the data sets.		
Validation Guidance: *Checks based on documentation:* • Check whether all information needed for the relevant legal and metrological purposes are contained in the data set. •		
Acceptable Solutions: • A legally and metrologically complete data set comprises the following fields: º Measurement value(s) with correct resolution º the legally correct unit of measure º the unit price or the price to pay (if applicable) º the place and time of the measurement (if applicable) º identification of the instrument if applicable (external storage) • Data are stored with the same resolution, values, units etc as indicated or printed on a delivery note.		

Additions for Risk Class E
Required Documentation (in addition to the documentation required for risk classes B, C and D)**:** Source code that generates the data sets for storing.
Validation Guidance (in addition to the guidance for risk classes B, C and D)**:** *Checks based on the source code:* • Check whether the data sets are built correctly.

Risk Class B	Risk Class C	Risk Class D

L2: Protection against accidental or unintentional changes
Stored data shall be protected against accidental and unintentional changes.

Specifying Notes:

1. Accidental changes of data can be caused by physical effects.
2. Unintentional changes are caused by the user of the device. Data housekeeping duties may require data belonging to paid-up or time-expired invoices to be deleted from time-to-time. Automatic or semi-automatic means should be used to ensure that only specified data is deleted and that the accidental deletion of “live” data is avoided. This is particularly important on networked systems and remote or removable storage where users might not realise the significance of the data.
3. A checksum shall be calculated by the receiver and compared with the attached nominal value. If the values match, the data set is valid and may be used; otherwise it must be deleted or marked invalid.

Required Documentation:

Description of protection measures (eg the checksum algorithm, including the length of the generator polynomial).

Required Documentation (in addition to the documentation required for risk classes B and C)**:**

The documentation shall show the measures taken to validate the effectiveness of the protection means.

Validation Guidance:

Checks based on documentation:

- Check that a checksum over data is generated.
- Check that legally relevant software, which reads the data and calculate a checksum really compares the calculated and the nominal values.
- Check that overwriting of data cannot occur before the end of the data storage period that is foreseen and documented by the manufacturer.
- Check that a warning is issued to the user if he is about to delete measurement data files.

Functional checks:

- Check by practical spot checks that before deleting measurement data a warning is given, if deleting is possible at all.

Validation Guidance (in addition to the guidance for risk classes B and C)**:**

Checks based on documentation:

- Check whether the measures taken are appropriate for the high protection level.

Acceptable Solutions:

- To detect data changes due to physical effects, a checksum with the **CRC-16** algorithm is calculated over the entire data set and inserted into the data set to be stored.

  Note: The algorithm is not secret and, in contrast to requirement L3, neither is the initial vector of the CRC-register nor the generator polynomial i.e. the devisor in the algorithm. The initial vector and generator polynomial are known to both of the programs that create and verify the checksums.
- Measurement data/invoice files could be protected by attaching an automatic date stamp on creation and a flag or label stating whether invoices were paid/unpaid. A utility program would only delete/move files if invoices had been paid or were out-of-date.
- Measurement data are not deleted without prior authorisation, e.g. a dialogue statement or window asking for confirmation of deletion.

Additions for Risk Class E

Required Documentation (in addition to the documentation required for risk classes B, C and D)**:**
Source code that realises the protection of stored data.

Validation Guidance (in addition to the guidance for risk classes B. C and D)**:**

Checks based on the source code:

- Check whether measures taken for protecting stored date are appropriate and correctly implemented.

<table>
<tr><th>Risk Class B</th><th>Risk Class C</th><th>Risk Class D</th></tr>
<tr><td colspan="3"><b>L3: Integrity of data</b>
<i>The measurement data stored must be protected against intentional changes. ~~carried out by simple common software tools.~~</i>
~~<b>L3: Integrity of data</b>~~
~~<i>The measurement data stored must be protected against intentional changes. carried out by special sophisticated software tools.. The protection level shall be equivalent to that required for electronic payment.</i>~~</td></tr>
<tr><td colspan="2"><b>Specifying Notes:</b>
1. This requirement applies to all types of storages except integrated storages.
2. The protection must apply against intentional changes carried out by simple common software tools.
3. Simple common software tools are understood as tools, which are easily available and manageable as e.g. office packages.</td><td><b>Specifying Notes:</b>
1. This requirement applies to all types of storages except integrated storages.
2. The protection must apply against intentional changes carried out by special sophisticated software tools.
3. “Sophisticated software tools” are for example debuggers, re-compilers, software development tools, etc.
4. The protection level shall be equivalent to that required for electronic payment.
5. Protection is realised by an electronic signature with an algorithm that guarantees that no identical signature results from different data sets.
<i>Note:</i> Even if the algorithm and key meet the level high, a technical solution with a standard personal computer would <b>not</b> realise this protection level provided that there are no appropriate protection means for the programs that sign or verify a data set (see basic guide <b>U</b> for universal computers, comment on requirement U6-D).</td></tr>
<tr><td colspan="2"><b>Required Documentation:</b>
The method of how the protection is realised shall be documented.</td><td><b>Required Documentation</b> (in addition to the documentation required for risk classes B and C)<b>:</b>
The protection measures taken shall be shown.</td></tr>
<tr><td colspan="2"><b>Validation Guidance:</b>
<i>Checks based on documentation:</i>
• If a checksum or signature is used
Check that the checksum or signature is generated over the entire data set.
Check that legally relevant software, which reads the data and calculate a checksum or decrypts a signature really compares calculated and the nominal values.
• Check that secret data (e.g. key initial value if used) are kept secret against spying out with simple tools.
<i>Functional checks:</i>
• Check that a falsified data set is rejected by the retrieval program.</td><td><b>Validation Guidance</b> (in addition to the guidance for risk classes B and C)<b>:</b>
<i>Checks based on documentation:</i>
• Check whether the measures taken are appropriate with respect to the required state of the art for a high protection level.</td></tr>
<tr><td colspan="2"><b>Acceptable Solutions:</b>
Just before the data is reused, the value of the checksum is recalculated and compared with the stored nominal value. If the values match, the data set is valid and may be used; otherwise it must be deleted or marked invalid.
An acceptable solution is the <b>CRC-16</b> algorithm.
<i>Note:</i> The algorithm is not secret but in contrast to requirement <b>L2,</b> the initial vector of the CRC-register or the generator polynomial (i.e. the divisor in the algorithm) must be. The initial vector and generator polynomial are known only to the programs generating and verifying the checksums. They must be treated as <i>keys</i> (see <b>L5</b>).</td><td><b>Acceptable Solutions:</b>
Instead of the CRC, a signature is calculated. A suitable signature algorithm would be one of the hash algorithms, e.g. SHA-1 or RipeMD160, in combination with an encryption algorithm such as RSA or Elliptic Curves. The minimum key length is 768 bits (RSA) or 128-160 bits (EC).</td></tr>
</table>

Additions for Risk Class E
Required Documentation (in addition to the documentation required for risk classes B and C)**:** Source code that realises the integrity of stored data.
Validation Guidance (in addition to the guidance for risk classes B and C)**:** *Checks based on the source code:* • Check whether measures taken for guaranteeing integrity are appropriate and correctly implemented.

<table>
<tr><th>Risk Class B</th><th>Risk Class C</th><th>Risk Class D</th></tr>
<tr><td colspan="3"><b>L4 Authenticity of measurement data stored</b>
<i>The measurement data stored must be capable of being authentically traced back to the measurement that generated them.</i></td></tr>
<tr><td colspan="3"><b>Specifying Notes:</b>
1. The authenticity of measurement data may be needed for reference at a later date, e.g., for checking invoices.
2. Authenticity requires the correct assignment (linking) of measurement data to the measurement that has generated the data.
3. Authenticity presupposes an identification of data sets.
4. Ensuring authenticity does not necessarily require an encryption of the data.</td></tr>
<tr><td colspan="2"><b>Required Documentation:</b>
Description of the method used for ensuring the authenticity.</td><td><b>Required Documentation</b> (in addition to the documentation required for risk classes B and C)<b>:</b>
The protection measures taken shall be shown.</td></tr>
<tr><td colspan="2"><b>Validation Guidance:</b>
<i>Checks based on documentation:</i>
• Check that there is a correct linking between each measurement value and the corresponding measurement.
• If a checksum or signature is used, check that the checksum or signature is generated over the entire data set.
• Check that secret data (e.g. key initial value if used) are kept secret against spying out with simple tools.
<i>Functional checks:</i>
• Check whether corresponding stored data and data printed on the ticket or invoice are identical.</td><td><b>Validation Guidance</b> (in addition to the guidance for risk classes B and C)<b>:</b>
<i>Checks based on documentation:</i>
• Check whether the measures taken are appropriate with respect to the required state of the art for a high protection level.</td></tr>
<tr><td colspan="3"><b>Acceptable Solutions:</b>
A stored data set contains the following data fields (additional to the fields defined in L3):
• A unique (current) identification number. The identification number is also copied to the delivery note.
• Time when the measurement has been performed (time stamp). The time stamp is also copied to the delivery note.
• An identification of the measuring instrument that has generated the value.
• A signature that is used for ensuring the integrity of data can simultaneously be used for ensuring the authenticity. The signature covers all of the fields of the data set. Refer to requirement L2, L3.
• The ticket may state that the measurement values can be compared with the reference data on a means of storage subject to legal control. Assignment is demonstrated by comparing the identification number or time stamp printed on the delivery note with that in the stored data set.</td></tr>
</table>

Additions for Risk Class E
Required Documentation (in addition to the documentation required for risk classes B and C)**:** Source code that generates the data sets for storing and realises the authentication..
Validation Guidance (in addition to the guidance for risk classes B and C)**:** *Checks based on the source code:* • Check whether the data sets are correctly built and reliably authenticated.

Risk Class B	Risk Class C	Risk Class D
~~**L5: Confidentiality of keys**~~ ~~*Keys must be treated as legally relevant data and must be kept secret and be protected against compromise by simple software tools*~~. **L5: Confidentiality of keys** *Keys and accompanying data must be treated as legally relevant data and must be kept secret and be protected against compromise by ~~sophisticated~~ software tools*. ~~*Appropriate methods equivalent to electronic payment shall be used. The user must be able to verify the authenticity of the public key.*~~		
Specifying Notes: 1. This requirement only applies if a secret key is used. 2. This requirement applies to measurement data storage, which are external from the measuring instrument or realised on universal computers. 3. The protection must apply against intentional changes carried out by common simple software tools. ~~3.~~4. If the access to the secret data is prevented, e.g., by sealing the housing of a built for purpose device, no additional software protection means are necessary.		**Specifying Notes:** 1. This requirement applies to storage in universal computers and to external storage. 2. The protection must apply against intentional changes carried out by special sophisticated software tools. 3. Appropriate methods equivalent to electronic payment shall be used. The user must be able to verify the authenticity of the public key.
Required Documentation: Description of the key management and means for keeping keys and associated information secret.		**Required Documentation** (in addition to the documentation required for risk classes B and C)**:** The protection measures taken shall be shown.
Validation Guidance: *Checks based on documentation:* • Check that the secret information cannot be compromised.		**Validation Guidance** (in addition to the guidance for risk classes B and C)**:** *Checks based on documentation:* • Check whether the measures taken are appropriate with respect to the required state of the art for a high protection level.
Acceptable Solutions: The secret key and accompanying data are stored in binary format in the executable code of the legally relevant software. It is then not obvious at which address these data are stored. The system software doesn't offer any features to view or edit these data. If the CRC algorithm is used as a signature, the initial vector or generator polynomial play the role of a key.		**Acceptable Solutions:** The secret key is stored in a hardware part that can be physically sealed. The software doesn't offer any features to view or edit these data. *Note:* A technical solution with a standard personal computer would not be sufficient to ensure high protection level if there were no appropriate hardware protection means for the key and other secret data (see basic guide for universal computers U6). 1) *Public Key Infrastructure:* The public key of the storage subject to legal control has been certified by an accredited Trust Centre. 2) *Direct Trust:* It is not necessary to involve a trust centre if, by prior agreement, both parties, are able to read the public key of the measuring instrument directly at a device subject to legal control that is displaying the relevant data set.

Additions for Risk Class E
Required Documentation (in addition to the documentation required for risk classes B and C)**:** Source code that realises key management.
Validation Guidance (in addition to the guidance for risk classes B and C)**:** *Checks based on the source code:* • Check whether measures taken for key management are appropriate.

Risk Class B	Risk Class C	Risk Class D
L6: Retrieval of stored data *The software used for verifying measurement data sets stored shall display or print the data, check the data for changes, and warn if a change has occurred. Data that are detected as having been corrupted must not be used.*		
Specifying Notes: 1. The measurement data stored might need to be referred to at a later date, e.g. transactions that are queried. If there is a doubt on the correctness of a delivery note or ticket, it must be possible to identify the measurement data stored to the disputed measurement without ambiguity refer also L1, L3, L4 and L5). 2. The identification number (see L1) must be printed out on the delivery note/ticket for the customer along with an explanation and a reference to the storage subject to legal control. 3. Verification means checking the integrity, authenticity and correct assignment of the measurement data stored. 4. The verification software used for displaying or printing the data stored shall be subject to legal control. 5. For instrument-specific requirements, refer to Extension I.		
Required Documentation: • Description of the functions of the retrieval program. • Description of detection of corruption. • Operating manual for this program.		**Required Documentation** (in addition to the documentation required for risk classes B and C)**:** The protection measures taken shall be shown.
Validation Guidance: *Checks based on documentation:* • Check that retrieval software really compares the calculated and the nominal values. • Check that retrieval software is part of the legally relevant software. *Functional checks:* • Check whether the program detects corrupted data sets. • Perform spot checks verifying that retrieval provides all necessary information.		**Validation Guidance** (in addition to the guidance for risk classes B and C)**:** *Checks based on documentation:* • Check whether the measures taken are appropriate with respect to the required state of the art for a high protection level.
Acceptable Solutions: The data set is read from the storage by the verifying program and the signature over all data fields is recalculated and compared with the stored nominal value. If both values match, the data set is correct, otherwise the data are not used and are deleted or marked invalid by the program.		

Additions for Risk Class E
Required Documentation (in addition to the documentation required for risk classes B and C)**:** Source code of the retrieval program.
Validation Guidance (in addition to the guidance for risk classes B and C)**:** *Checks based on the source code:* • Check whether measures taken for retrieval, verification of signatures etc. are appropriate and correctly implemented.

Risk Class B	Risk Class C	Risk Class D

L7: Automatic storing
The measurement data must be stored automatically when the measurement is concluded.

Specifying Notes:

1. This requirement applies to all types of storage.
2. This requirement means that the storing function must not depend on the decision of the operator. Nevertheless, in some types of instrument, e.g. weighing instruments, a decision or command is required from the operator whether or not to accept the result. In other words, there might be some intermediate measurements that will not be stored (for example during loading or before the quantity of product requested is on the load receptor). However, even in this case, the result will be stored automatically when the operator accepts the result.
3. For the case of full storage, refer to requirement L8.

Required Documentation:
Confirmation that storing is automatically carried out. Description of the Graphical User Interface.

Validation Guidance:

Functional checks:

- Examine by spot checks that the measurement values are stored automatically after measurement or acceptance of measurement is concluded. Check that there are no buttons or menu items to interrupt or disable the automatic storing.

Acceptable Solutions:
There is no menu item or button in the Graphical User Interface (GUI) that supports manual initiation of storing measurement results. The measurement values are wrapped in a data set along with additional information such as time stamp and signature and are stored immediately after the measurement, or the acceptance of measurement, respectively.

Additions for Risk Class E

Required Documentation (in addition to the documentation required for risk classes B, C and D)**:**
Source code of the instrument.

Validation Guidance (in addition to the guidance for risk classes B, C and D*)***:**

Checks based on the source code:

- Check whether measures taken for automatic storing are appropriate and correctly implemeted.

Risk Class B	Risk Class C	Risk Class D

L8: Storage capacity and continuity

The long-term storage must have a capacity which is sufficient for the intended purpose.

Specifying Notes:

1. When an storage is full or removed/disconnected from the instrument, a warning shall be given to the operator. For further necessary actions refer to the measuring instrument-specific requirements (Extension I).
2. The regulation concerning the minimum period for storing measurement data is beyond the scope of this requirement and is left to national regulations. It is the responsibility of the owner to have an instrument with sufficient storage capacity to fulfil the requirements applicable to his activity. The notified body for EC type examination will check only that the data are stored and retrieved correctly and whether new transactions are inhibited when the storage is full.
3. It is also beyond the scope of this requirement to require certain inscriptions on the device as concerning the capacity of the storage the capacity or other accompanying information that allow to calculate the capacity. However, the manufacturer shall make available the information on the capacity.

Required Documentation:

Description of management of exceptional cases when storing measurement values.

Validation Guidance:

Checks based on documentation:

- Check that the capacity of storage or a formula for calculating it is given by manufacturer.
- Check that overwriting of data cannot occur before the end of the data storage period that is foreseen and documented by the manufacturer.

Functional checks:

- Check that a warning is issued to the user if he is about to delete measurement data files (if deleting is possible at all).
- Check that a warning is given if the storage is full or removed.

Acceptable Solutions:

- For interruptible measurements that can be stopped easily and rapidly, e.g. weighing, fuel measurement, etc, the measurement may be completed even if the storage becomes unavailable. The measuring instrument or the device should have a buffer that is large enough to store the current transaction. After this, no new transaction may be started and the buffered values are kept for later transmission to a fresh storage.
- Measurements that are not interruptible, e.g. the measurement of energy, volume, etc, do not need a special intermediate buffer because these measurements always are cumulative. The cumulative register can be read out and transmitted to the storage at a later time when the storage is available again.
- Measurement data may be automatically overwritten by a utility that checks if the measurement data is out-of-date (refer to national regulations for the relevant time period) or that the invoice has been paid. The utility shall prompt the user for permission to delete and data shall be deleted in the order oldest first.)

Additions for Risk Class E

Required Documentation (in addition to the documentation required for risk classes B, C and D)**:**

Source code that realises storing of data.

Validation Guidance (in addition to the guidance for risk classes B, C and D)**:**

Checks based on the source code:

- Check whether measures taken for storing are appropriate and correctly implemented.

7 Extension T: Transmission of Measurement Data via Communication Networks

This is an extension to the software requirements of the basic guides P and U. It must be used only if measurement data are transmitted via communication networks to a distant device where they are further processed and/or used for legally regulated purposes. This extension dos not apply if there is no subsequent legally relevant data processing. If software is downloaded to a device subject to legal control the requirements of Extension D apply.

7.1 Technical description

The set of requirements of this extension applies only if the device under consideration is connected to a network and transmits or receives measurement data that are legally relevant. In the following table three network configurations are identified. The simplest is an array of devices that are all subject to legal control. The participants are fixed at legal verification. A variant to this (closed network, partly under legal control), is a net with participants that are not subject to legal control but all are known and do not change during operation. An *open network* has no limitation in identity, functionality, presence and location of the participants.

Description of configurations
Closed network, completely under legal control Only a fixed number of participants with clear identity, functionality and location are connected. All devices are subject to legal control. No devices exist in the network that are not subject to legal control.
Closed network, partly under legal control A fixed number of participants with clear identity and location are connected to the network. Not all devices are subject to legal control and therefore their functionality is unknown.
Open network Arbitrary participants (devices with arbitrary functions) can connect to the network. The identity and functionality of a participating device and its location may be unknown to other participants. Any network that contains legally controlled devices with IR or wireless network communications interfaces shall be considered to be an open network.

7.2 Specific software Requirements for Data Transmission

<table>
<tr><th>Risk Class B</th><th>Risk Class C</th><th>Risk Class D</th></tr>
<tr><td colspan="3"><b>T1: Completeness of transmitted data</b>
<i>The transmitted data must contain all relevant information necessary to present or further process the measurement result in the receiving unit.</i></td></tr>
<tr><td colspan="3"><b>Specifying Notes:</b>
1. The metrological part of a transmitted data set comprises one or more measurement values with correct resolution, the legally correct unit of measure and depending on the application the unit price or the price to pay and the place of the measurement.</td></tr>
<tr><td colspan="3"><b>Required Documentation:</b>
Document all fields of the data set.</td></tr>
<tr><td colspan="3"><b>Validation Guidance:</b>
<i>Checks based on documentation:</i>
• Check whether all information for further processing the measurement values at the receiving unit are contained in the data set.</td></tr>
<tr><td colspan="3"><b>Acceptable Solutions:</b>
The data set comprises the following fields:
• Measurement value(s) with correct resolution
• the legally correct unit of measure
• the unit price or the price to pay (if applicable)
• the time and date of the measurement (if applicable)
• identification of the instrument if applicable (data transmission)
• the place of the measurement (if applicable)</td></tr>
</table>

<table>
<tr><th>Additions for Risk Class E</th></tr>
<tr><td><b>Required Documentation</b> (in addition to the documentation required for risk classes B, C and D)<b>:</b>
Source code that generates the data sets for transmission.</td></tr>
<tr><td><b>Validation Guidance</b> (in addition to the guidance for risk classes B, C and D)<b>:</b>
<i>Checks based on the source code:</i>
• Check whether data sets are built correctly.</td></tr>
</table>

<table>
<tr><th>Risk Class B</th><th>Risk Class C</th><th>Risk Class D</th></tr>
<tr><td colspan="3"><b>T2: Protection against accidental or unintentional changes</b>
<i>Transmitted data shall be protected against accidental and unintentional changes.</i></td></tr>
<tr><td colspan="3"><b>Specifying Notes:</b>
1. Accidental changes of data can be caused by physical effects.
2. Unintentional changes are caused by the user of the device.
3. Means shall be provided to detect transmission errors.~~A checksum shall be calculated by the receiver and compared with the attached nominal value. If the values match, the data set is valid and may be used; otherwise it must be deleted or marked invalid.~~</td></tr>
<tr><td colspan="2"><b>Required Documentation:</b>
Description of the checksum algorithm, if used, including the length of the generator polynomial.
Description of an alternative method if used.</td><td><b>Required Documentation</b> (in addition to the documentation required for risk classes B and C)<b>:</b>
The documentation shall show the measures taken to validate the effectiveness of the protection means.</td></tr>
<tr><td colspan="2"><b>Validation Guidance:</b>
<i>Checks based on documentation:</i>
• Check that a checksum over data is generated.
• Check that legally relevant software that receives the data re-calculates the checksum and compares it with the nominal value contained in the data set.</td><td><b>Validation Guidance</b> (in addition to the guidance for risk classes B and C)<b>:</b>
<i>Checks based on documentation:</i>
• Check whether the measures taken are appropriate for the high protection level.</td></tr>
<tr><td colspan="3"><b>Acceptable Solutions:</b>
1) To detect data changes, a checksum with the <b>CRC-16</b> algorithm is calculated over all bytes of a data set and inserted into the data set to be transmitted. Just before the data is reused, the value of the checksum is recalculated by the receiver and compared with the attached nominal value.

If the values match, the data set is valid and may be used, otherwise it must be deleted or marked invalid.
<i>Note:</i> The algorithm is not secret and, in contrast to requirement T3, neither is the initial vector of the CRC-register nor the generator polynomial i.e. the devisor in the algorithm. The initial vector and generator polynomial are known to both of the programs that create and verify the checksums.
2) Use of means provided by transmission protocols e.g. TCP/IP, IFSF.</td></tr>
</table>

Additions for Risk Class E
Required Documentation (in addition to the documentation required for risk classes B, C and D)**:** Source code that realises the protection of transmitted data.
Validation Guidance (in addition to the guidance for risk classes B, C and D)**:** *Checks based on the source code:* • Check whether measures taken for protecting transmitted data are appropriate.

Risk Class B	Risk Class C	Risk Class D
T3: Integrity of data *The legally relevant transmitted data must be protected against intentional changes with ~~simple common~~ software tools.* ~~**T3: Integrity of data**~~ ~~*The legally relevant transmitted data must be protected against intentional changes with special sophisticated software tools. The protection level shall be equivalent to that required for electronic payment.*~~		
Specifying Notes: 1. This requirement only applies to networks that are open or only partly under legal control, not to closed networks. 2. The protection must apply against intentional changes carried out by common simple software tools. ~~2.~~3. Simple common software tools are understood as tools, which are easily available and manageable as e.g. office packages		**Specifying Notes:** 1. This requirement applies to open networks and to closed networks partly under legal control. 2. Protection is realised by an electronic signature with an algorithm that guarantees that no identical signature results from different data sets. 3. The protection must apply against intentional changes carried out by special sophisticated software tools. ~~3.~~4. "Sophisticated software tools" are eg. debuggers, re-compilers, software developing tools, etc.~~,~~ 5. The protection level shall be equivalent to that required for electronic payment. *Note:* Even if the algorithm and key meet the level high, a technical solution with a standard personal computer would **not** realise this protection level provided that there are no appropriate protection means for the programs that sign or verify a data set (see basic guide **U** for universal computers, comment on requirement U6-D).
Required Documentation: Description of the protection method		**Required Documentation** (in addition to the documentation required for risk classes B and C)**:** The protection measures taken shall be shown.
Validation Guidance: *Checks based on documentation:* • If a checksum or signature is used: Check that the checksum or signature is generated over the entire data set. Check that legally relevant software that receives the data re-calculates the checksum or decrypts the signature and compares it with the nominal value contained in the data set. • Check that secret data (e.g. key initial value if used) are kept secret against spying out with simple tools.		**Validation Guidance** (in addition to the guidance for risk classes B and C)**:** *Checks based on documentation:* • Check whether the measures taken are appropriate with respect to the required state of the art for a high protection level.
Acceptable Solutions: • A checksum is generated of the data set to be transmitted. Just before the data is reused, the value of the checksum is recalculated and compared with the nominal value that is contained in the received data set. If the values match, the data set is valid and may be used, otherwise it must be deleted or marked invalid. • An acceptable solution is the **CRC-16** algorithm. *Note:* The algorithm is not secret but in contrast to requirement **L2,** the initial vector of the CRC-register or the generator polynomial (i.e. the divisor in the algorithm) are secret. The initial vector and generator polynomial are known only to the programs generating and verifying the checksums. They must be treated as *keys* (see **L5**).		**Acceptable Solutions:** • Instead of the CRC a signature is calculated. A suitable signature algorithm would be e.g. one of the hash algorithms SHA-1 or RipeMD-160 in combination with an encryption algorithm like RSA or Elliptic Curves. The minimum key length is 768 bits (RSA) or 128-160 bits (EC). • Protection is provided by some transmission protocols e.g. HTTPS

Additions for Risk Class E
Required Documentation (in addition to the documentation required for risk classes B and C)**:** Source code that realises the integrity of transmitted data.
Validation Guidance (in addition to the guidance for risk classes B and C)**:** *Checks based on the source code:* • Check whether measures taken for guaranteeing integrity of transmitted data are appropriate.

<table>
<tr><th>Risk Class B</th><th>Risk Class C</th><th>Risk Class D</th></tr>
<tr><td colspan="3"><b>T4: Authenticity of transmitted data</b>
<i>For the receiving program of transmitted relevant data, it shall be possible to verify the authenticity and the assignment of measurement values to a certain measurement.</i></td></tr>
<tr><td colspan="3"><b>Specifying Notes:</b>
1a In a network with unknown participants, it is necessary to identify the origin of measurement data transmitted without ambiguity. (The authenticity relies on the identification number of the data set and the network address).
1b In a closed network all participants are known. No additional IT means are necessary, but the topology of the network (the number of participants) shall be fixed by sealing.
2. Unforeseen delays are possible during transmission. For a correct assignment of a received measurement value to a certain measurement the time of measurement must be registered.
3. To ensure the authenticity, an encryption of measurement data is not necessarily required.</td></tr>
<tr><td colspan="2"><b>Required Documentation:</b>
<i>Network with unknown participants:</i> Description of the IT means for correct assigning of measurement value to measurement.
<i>Closed network:</i> Description of the hardware means preserving the number of participants in the network. Description of initial identification of the participants.</td><td><b>Required Documentation</b> (in addition to the documentation required for risk classes B and C)<b>:</b>
The protection measures taken shall be shown.</td></tr>
<tr><td colspan="2"><b>Validation Guidance:</b>
<i>Checks based on documentation:</i>
• Check that there is a correct linking between each measurement value and the corresponding measurement.
• Check that data are digitally signed to ensure their proper identification and authentication.</td><td><b>Validation Guidance</b> (in addition to the guidance for risk classes B and C)<b>:</b>
<i>Checks based on documentation:</i>
• Check whether the measures taken are appropriate with respect to the required state of the art for a high protection level.</td></tr>
<tr><td colspan="3"><b>Acceptable Solutions:</b>
• Each data set has a unique (current) identification number, which may contain the time when the measurement has been performed (time stamp).
• Each data set contains information about the origin of the measurement data, i.e. serial number or identity of the measuring instrument that generated the value.
• In a network with unknown participants, authenticity is guaranteed if the data set carries an unambiguous signature. The signature covers all of these fields of the data set
• The receiver of the data set checks all data for plausibility.</td></tr>
</table>

Additions for Risk Class E
Required Documentation (in addition to the documentation required for risk classes B and C)**:** Source code of sending and receiving device..
Validation Guidance (in addition to the guidance for risk classes B and C)**:** *Checks based on the source code:* • Check whether measures taken for guaranteeing the authenticity of transmitted data are appropriate.

<table>
<tr><th>Risk Class B</th><th>Risk Class C</th><th>Risk Class D</th></tr>
<tr><td colspan="3">~~<b>T5: Confidentiality of keys</b>~~
~~<i>Keys must be treated as legally relevant data and must be kept secret and be protected against compromise by simple software tools.</i>~~
<b>T5: Confidentiality of keys</b>
<i>Keys and accompanying data must be treated as legally relevant data and must be kept secret and be protected against compromise by ~~sophisticated~~ software tools. ~~Appropriate methods equivalent to electronic payment shall be used.~~</i></td></tr>
<tr><td colspan="2"><b>Specifying Notes:</b>
1. This requirement only applies if a secret key exists in the system. (Normally not in Closed networks.).
2. The protection must apply against intentional changes carried out by common simple software tools.
~~2.~~3. If the access to the secret data is prevented e.g. by sealing the housing of a built for purpose device, no additional software protection means are necessary.</td><td><b>Specifying Notes:</b>
1. This requirement only applies if a secret key exists in the system. (Normally not in Closed networks.)
2. The protection must apply against intentional changes carried out by special sophisticated software tools.
~~2.~~3. The received measurement values are read from the data set and their signature is checked with the aid of the public key of the sending measuring instrument (or the device that generated the relevant data set). With this check the receiver can prove that value and signature belong together.
4. Appropriate methods equivalent to electronic payment shall be used.</td></tr>
<tr><td colspan="2"><b>Required Documentation:</b>
Description of the key management and means for keeping keys and associated information secret.</td><td><b>Required Documentation</b> (in addition to the documentation required for risk classes B and C)<b>:</b>
The protection measures taken shall be shown.</td></tr>
<tr><td colspan="2"><b>Validation Guidance:</b>
<i>Checks based on documentation:</i>
• Check that the secret information cannot be compromised.</td><td><b>Validation Guidance</b> (in addition to the guidance for risk classes B and C)<b>:</b>
<i>Checks based on documentation:</i>
• Check whether the measures taken are appropriate with respect to the required state of the art for a high protection level.</td></tr>
<tr><td colspan="2"><b>Acceptable Solutions:</b>
The secret key and accompanying data are stored in binary format in the executable code of the legally relevant software. It is then not obvious at which address these data are stored. The system software doesn't offer any features to view or edit these data. If the CRC algorithm is used as a signature, the initial vector or generator polynomial play the role of a key.</td><td><b>Acceptable Solutions:</b>
The secret key is stored in a hardware part that can be physically sealed. The software doesn't offer any features to view or edit these data.
<u>Note:</u> A technical solution with a standard personal computer would not be sufficient to ensure high protection level if there were no appropriate hardware protection means for the key and other secret data (see basic guide for universal computers U6).
1) <i>Public Key Infrastructure:</i> The public key of the storage subject to legal control has been certified by an accredited Trust Centre.
2) <i>Direct Trust:</i> It is not necessary to involve a trust centre if, by prior agreement, both parties, are able to read the public key of the measuring instrument directly at a device subject to legal control that is displaying the relevant data set.</td></tr>
</table>

Additions for Risk Class E
Required Documentation (in addition to the documentation required for risk classes B and C)**:** Source code that realises key management.

Validation Guidance (in addition to the guidance for risk classes B and C)**:** *Checks based on the source code:* • Check whether measures taken for key management are appropriate.

Risk Class B	Risk Class C	Risk Class D
T6: Handling of corrupted data *Data that are detected as having been corrupted must not be used.*		
Specifying Notes: 1. Though communication protocols normally repeat transmission until it succeeds, it nevertheless is possible that a corrupted data set is received.		
Required Documentation: Description of the detection of transmission faults or intentional changes..		**Required Documentation** (in addition to the documentation required for risk classes B and C)**:** The measures taken for correct handling of corrupted data shall be shown.
Validation Guidance: *Checks based on documentation and functional checks:* • Check that the corrupted data will not be used according to the delivered concept		**Validation Guidance** (in addition to the guidance for risk classes B and C)**:** *Checks based on documentation:* • Check whether the measures taken are appropriate with respect to the required state of the art for a high protection level.
Acceptable Solutions: When the program that is receiving data sets detects a discrepancy between the data set and the nominal value of the signature, it first tries to reconstruct the original value if redundant information is available. If reconstruction fails, it generates a warning to the user, does not output the measurement value and • Sets a flag in a special field of the data set (status field) with the meaning “not valid” OR • Deletes the corrupted data set.		

Additions for Risk Class E
Required Documentation (in addition to the documentation required for risk classes B and C)**:** Source code of the receiving device.
Validation Guidance (in addition to the guidance for risk classes B and C)**:** *Checks based on the source code:* • Check whether measures taken for handling corrupted data are appropriate.

Risk Class B	Risk Class C	Risk Class D
T7: Transmission delay *The measurement must not be inadmissibly influenced by a transmission delay.*		
Specifying Notes: The manufacturer shall investigate the timing of the data transmission and shall guarantee that under worst case conditions the measurement is not inadmissibly influenced.		
Required Documentation: Description of the concept, how measurement is protected against transmission delay.		
Validation Guidance: • Check the concept that the measurement is not influenced by transmission delay.		
Acceptable Solutions: Implementation of transmission protocols for field buses.		

Additions for Risk Class E

Required Documentation (in addition to the documentation required for risk classes B, C and D)**:**
Source code that realises the data transmission.

Validation Guidance (in addition to the guidance for risk classes B, C and D)**:**

Checks based on the source code:

- Check whether measures taken for handling transmission delay are appropriate.

Risk Class B	Risk Class C	Risk Class D

T8: Availability of transmission services
If network services become unavailable, no measurement data must get lost.

Specifying Notes:

1. The user of the measuring system must not be able to corrupt measurement data by suppressing transmission.
2. Transmission disturbances happen accidentally and cannot be excluded. The sending device must be able to handle this situation.
3. The reaction of the instrument if transmission services become unavailable depends on the measuring principle (see Part I).

Required Documentation:
Description of protection measures against transmission interruption or other failures.

Validation Guidance:

Checks based on documentation:

- Check by what measures are implemented to protect from data loss.
- Check which measures are foreseen for the case of transmission failure.

Functional checks:

- Spot checks shall show that no relevant data get lost due to a transmission interruption.

Acceptable Solutions:

1) For interruptible measurements that can be stopped easily and rapidly, e.g. weighing, fuel measurement, etc, the measurement may be completed even though the transmission is down. However, the measuring instrument or the device that is transmitting the legally relevant data must have a buffer that is large enough to store the current transaction. After this no new transaction may be started and the buffered values are kept for later transmission. For other examples see part I.
2) Measurements that are not interruptible, e.g. the measurement of energy, volume, etc, do not need a special intermediate buffer because these measurements always are cumulative. The cumulative register can be read out and transmitted at a later time when the connection is up again.

Additions for Risk Class E

Required Documentation (in addition to the documentation required for risk classes B, C and D)**:**
Source code that realises data transmission.

Validation Guidance (in addition to the guidance for risk classes B, C and D)**:**

Checks based on the source code:

- Check whether measures taken for reacting on interrupted transmission service are appropriate.

8 Extension S: Software Separation

Software separation is an optional design methodology that allows the manufacturer to easily modify non-legally relevant software. If software separation is implemented, then this extension shall be considered in addition to the basic requirements for types P and U.

8.1 Technical description

Software controlled measuring instruments or systems in general have complex functionality and contain modules that are legally relevant and modules that are not. It is advantageous for the manufacturer and examiner – though it is not prescribed – to separate these software modules of the measuring system.

In the following table, two variants of software separation are described. Both variants are covered by the set of requirements.

Description
Software separation is realised independently from the operating system within an application domain, i.e., at the *programming language level (**Low level software separation**).* *Note:* This feature is realisable in both built-for-purpose devices and universal computers.
The software modules to be separated are realised as independent objects in terms of the *operating system (**High level software separation**).* *Note:* This type of separation is normally possible only with universal computers. Example solutions are independently executable programs, dynamically linked libraries etc.

The protection against inadmissible changes of measurement values and parameters is only addressed indirectly as the programmer of software parts that are not subject to legal control must not give the user of the measuring system the opportunity of corruption. But this has in any case to be considered by the programmer (with or without separation) and the appropriate requirements are given in the basic parts P and U (chapter 4 and 5) of the guide.

8.2 Specific software requirements for software separation

Risk Class B	Risk Class C	Risk Class D
S1: Realisation of software separation *There shall be a part of the software that contains all legally relevant software and parameters that is clearly separated from other parts of software.~~ that is subject to legal control shall contain all legally relevant software and parameters.~~*		
Specifying Notes: 1. In the case of *low level separation,* all *program units* (subroutines, procedures, functions, classes, etc.) and in case of *high level separation* all *programs and libraries* ° that contribute to the calculation of measurement values or have an impact on it, ° that contribute to auxiliary functions such as displaying data, data security, data storage, software identification, performing software download, data transmission or storing, verifying received or stored data etc. belong to the legally relevant software. 2. All *variables, temporary files and parameters* that have an impact on measurement value or on legally relevant functions or data belong to the legally relevant software. 3. The components of a protective software interface (see S~~E3.~~3) are part of the legally relevant software. 4. Non-legally relevant software comprises the remaining program units, data or parameters not covered above. Modifications to this part are allowed without informing the NB provided the subsequent requirements of software separation are observed.		
Required Documentation: Description of all components described in the specifying notes above that belong to the legally relevant software.		**Required Documentation** (in addition to the documentation required for risk classes B and C)**:** The correct implementation of software separation shall be shown by the documentation.
Validation Guidance: *Checks based on documentation:* • Check that all legally relevant components mentioned in specifying notes 1 through 3 are included in legally relevant software.		**Validation Guidance** (in addition to the guidance for risk classes B and C)**:** *Checks based on documentation:* • Check whether the realisation of software separation is correct.
Acceptable Solutions: As described by the requirement itself.		

Additions for Risk Class E
Required Documentation (in addition to the documentation required for risk classes B and C)**:** Source code of the legally relevant software.
Validation Guidance (in addition to the guidance for risk classes B and C)**:** *Checks based on the source code:* • Check the software design whether data flow concerning legally relevant data is unambiguously defined in the legally relevant software and can be verified. • Check (eg by data flow analysis with tools or manually) that all program units, programs or libraries that are involved in processing the measurement values are registered to the legally relevant software. • Search inadmissible data flow from parts not subject to legal control to domains to be protected.

Risk Class B	Risk Class C	Risk Class D
S2: Mixed indication *Additional information generated by the software, which is not legally relevant, may only be shown on a display or printout, if it cannot be confused with the information that originates from the legally relevant part.*		
Specifying Notes: As the programmer of the non-legally relevant software may not know about the admissibility of indications, it is the responsibility of the manufacturer to guarantee that all indicated information fulfil the requirement.		
Required Documentation: Description of the software that realises the indication. Description how the indication of legally relevant information is		**Required Documentation** (in addition to the documentation required for risk classes B and C)**:** The realisation of mixed indication shall be shown by the documentation.
Validation Guidance: *Functional checks:* • Judge through visual check that additional information generated by non-legally relevant software and presented on display or printout can not be confused with the information originating from legally relevant software.		**Validation Guidance** (in addition to the guidance for risk classes B and C)**:** *Checks based on documentation:* • Check whether the realised implementation of mixed indication is correct.
Acceptable Solutions: • The information to be displayed by the non-legally relevant software is transferred via the protective interface (see E3.3) to the legally relevant software. Behind the interface, it passes through a filter that detects inadmissible information. The admissible information is then inserted into the indication controlled by the legally relevant software. • On a window-style display (universal computer) the legally relevant software checks in short time intervals whether the window with the legally relevant information is always visible and on top of the window stack. If it is hidden, minimised or outside the border, the software generates a warning or stops the output and processing of measurement values. When the measurement is finished, the window for legal purposes may be closed.		

Additions for Risk Class E
Required Documentation (in addition to the documentation required for risk classes B and C)**:** Source code of the legally relevant software.
Validation Guidance (in addition to the guidance for risk classes B and C)**:** *Checks based on the source code:* • Check that legally relevant software generates the indication of measurement values. • Check that this indication cannot be changed or suppressed by non-legally relevant programs.

<table>
<tr><th>Risk Class B</th><th>Risk Class C</th><th>Risk Class D</th></tr>
<tr><td colspan="3"><b>S3: Protective software interface</b>
<i>The data exchange between the legally relevant and non-legally relevant software must be performed via a protective software interface, which comprises the interactions and data flow.</i></td></tr>
<tr><td colspan="3"><b>Specifying Notes:</b>
1. All interactions and data flows shall not inadmissibly influence the legally relevant software.
2. There shall be an unambiguous assignment of each command sent via the software interface to the initiated function or data change in the legally relevant software.
3. Codes and data that are not declared and documented as commands must not have any effect on the legally relevant software.
4. The interface shall be completely documented and any other non-documented interaction or data flow (circumvention of the interface) must not be realised neither by the programmer of the legally relevant software nor by the programmers of the non-legally relevant software.
<u>Note:</u> The programmers are responsible for observing these constraints. Technical means to prevent them from circumventing the software interface are not possible. The programmer of the protective interface should be instructed about this requirement.</td></tr>
<tr><td colspan="2"><b>Required Documentation:</b>
• Description of the software interface, especially which data domains realise the interface.
• A complete list of all commands together with a declaration of completeness.
• A brief description of their meaning and their effect on the functions and data of the measuring instrument.</td><td><b>Required Documentation</b> (in addition to the documentation required for risk classes B and C)<b>:</b>
The realisation of the software interface shall be shown by the documentation.</td></tr>
<tr><td colspan="2"><b>Validation Guidance:</b>
<i>Checks based on documentation:</i>
• Check that functions of the legally relevant software, that may be triggered via the protective software interface are defined and described.
• Check that the parameters that may be exchanged via the interface are defined and described.
• Check that the description of the functions and parameters is conclusive and complete.</td><td><b>Validation Guidance</b> (in addition to the guidance for risk classes B and C)<b>:</b>
<i>Checks based on documentation:</i>
• Check whether realisation of the software interface is correct.</td></tr>
<tr><td colspan="3"><b>Acceptable Solutions:</b>
• The data domains of the legally relevant software part are encapsulated by declaring only local variables in the legally relevant part.
• The interface is realised as a subroutine belonging to the legally relevant software that is called from the non-legally relevant software. The data to be transferred to the legally relevant software are passed as parameters of the subroutine.
• The legally relevant software filters out inadmissible interface commands.</td></tr>
</table>

<table>
<tr><th>Additions for Risk Class E</th></tr>
<tr><td><b>Required Documentation</b> (in addition to the documentation required for risk classes B and C)<b>:</b>
Source code of the legally relevant software.</td></tr>
<tr><td><b>Validation Guidance</b> (in addition to the guidance for risk classes B and C)<b>:</b>
<i>Checks based on the source code:</i>
• Check the software design whether data flow is unambiguously defined in the legally relevant software and can be verified.
• Check the data flow via the software interface with tools or manually. Check whether all data flow between the parts has been documented (no circumvention of the declared software interface).
• Search inadmissible data flow from the part not subject to control to domains to be protected.
• Check that commands, if any, are decoded correctly and no undocumented commands exist.</td></tr>
</table>

9 Extension D: Download of Legally Relevant Software

This extension shall be used for the download of legally relevant software, e.g. bug-fixes, updates, new applications, etc to measuring instruments of both types, P and U, as appropriate. These requirements are to be considered in addition to the basic requirements for Types P and Type-U described in chapters 4 and 5 in the guide.

9.1 Technical Description

Software may be downloaded only to measuring instruments that are characterised by the following properties:

Hardware Configuration
The target device is subject to legal control. It may be a built-for-purpose measuring instrument (Type P) or one based on a universal computer (Type U). Communications links for the download may be direct, e,g, RS 232, USB, over a closed network partly or wholly under legal control, e.g. Ethernet, token-ring LAN, or over an open network, e.g. Internet.
Software Configuration
The entire software of the target device may be legally controlled or it may have software separation. The download of legally relevant software must follow the requirements outlined below. If there is no software separation in the measuring instrument, then all of the requirements below apply to all downloads.

9.2 Specific Software Requirements

<table>
<tr><th>Risk Class B</th><th>Risk Class C</th><th>Risk Class D</th></tr>
<tr><td colspan="3"><b>D1: Download mechanism</b>
<i>Downloading and the subsequent installation of software shall be automatic and shall ensure that the software protection environment is at the approved level on completion.</i></td></tr>
<tr><td colspan="3"><b>Specifying Notes:</b>
1. Downloading shall be automatic to ensure that the existing level of protection is not compromised.
2. The target device has a fixed legally relevant software that contains all of the checking functions necessary for fulfilling requirements D2 to D5.
3. The instrument should be capable of detecting if the download or installation fails. A warning shall be given. If the download or installation is unsuccessful or is interrupted, the original status of the measuring instrument shall be unaffected. Alternatively, the instrument shall display a permanent error message and its metrological functioning shall be inhibited until the cause of the error is corrected.
4. On successful completion of the installation, all protective means should be restored to their original state unless the downloaded software has NB authorisation in the TAC to amend them.
5. During download and the subsequent installation of downloaded software, measurement by the instrument shall be inhibited or correct measurement shall be guaranteed.
6. The fault handling requirements described in Extension I may be implemented if faults occur during downloading. The number of re-installation attempts shall be limited.
7. If the requirements D2 to D5 cannot be fulfilled, it is still possible to download the non-legally relevant software part. In this case the following requirements shall be met:
• There is a distinct separation between the legally relevant and non-relevant software according to Extension S.
• The whole legally relevant software part is fixed i.e. it cannot be downloaded or changed without breaking a seal.
• It is stated in the TAC downloading of the non-legally relevant part is acceptable.</td></tr>
<tr><td colspan="2"><b>Required Documentation:</b>
The documentation should briefly describe the automatic nature of the download, checking, installation, how the level of protection is guaranteed on completion, what happens if a fault occurs.</td><td><b>Required Documentation</b> (in addition to the documentation required for risk classes B and C)<b>:</b>
The realisation of the download mechanism shall be shown by the documentation.</td></tr>
<tr><td colspan="2"><b>Validation Guidance:</b>
<i>Checks based on documentation:</i>
• Check the documentation how the download procedure is managed.
• Check that downloading and installation is handled automatically, that the measuring instrument is locked (if appropriate) and that software protection is not compromised following a download.
• Check that there exists non-downloadable fixed legally relevant software for authenticity and integrity checks.
• Check that during software download no measurement is possible or correct measurement is guaranteed.
<i>Functional checks:</i>
• Perform at least one software download to check the correct software download.</td><td><b>Validation Guidance</b> (in addition to the guidance for risk classes B and C)<b>:</b>
<i>Checks based on documentation:</i>
• Check whether the realisation of the download mechanism is correct.</td></tr>
<tr><td colspan="3"><b>Acceptable Solutions:</b>
A utility program resident in the fixed part of the software that:
a. Handshakes with the sender and checks for consent
b. Automatically inhibits measurement unless correct measurement can be guaranteed
c. Automatically downloads the legally relevant software to a secure holding area
d. Automatically carries out the checks required by D2 to D4
e. Automatically installs the software into the correct location
f. Takes care of housekeeping, e.g. deletes redundant files, etc.
g. Ensures that any protection removed to facilitate downloading and installation is automatically replaced to the approved level on completion.
h. Initiates the appropriate fault handling procedures if a fault occurs</td></tr>
</table>

Additions for Risk Class E
Required Documentation (in addition to the documentation required for risk classes B and C)**:** Source code of the fixed software part responsible for the management of the download process..
Validation Guidance (in addition to the guidance for risk classes B and C)**:** *Checks based on the source code:* • Check whether measures taken for managing the download process are appropriate.

<table>
<tr><th>Risk Class B</th><th>Risk Class C</th><th>Risk Class D</th></tr>
<tr><td colspan="3"><b>D2: Authentication of downloaded software</b>
<i>Means shall be employed to guarantee that the downloaded software is authentic, and to indicate that the downloaded software has been approved by an NB.</i></td></tr>
<tr><td colspan="3"><b>Specifying Notes:</b>
1. Before the downloaded software is used for the first time, the measuring instrument shall automatically check that:
a. The software is authentic (not a fraudulent simulation).
b. The software is approved for that type of measuring instrument.
2. The means by which the software identifies its NB approval status shall be made secure to prevent counterfeiting of the NB status.
3. If downloaded software fails any of the above tests, see D1.</td></tr>
<tr><td colspan="2"><b>Required Documentation:</b>
The documentation should describe:
• How authenticity of the software identification is guaranteed.
• How the authenticity of NB approval is guaranteed.
• How it is guaranteed that the downloaded software is approved for the type of measuring instrument to which it has been downloaded.</td><td><b>Required Documentation</b> (in addition to the documentation required for risk classes B and CZ<b>:</b>
The realisation of authentication shall be shown by the documentation.</td></tr>
<tr><td colspan="2"><b>Validation Guidance:</b>
<i>Checks based on documentation and functional checks:</i>
• Check the documentation, how a download of fraudulent software is prevented.
• Check through functional tests that a download of fraudulent software is prevented.
Ensure the authentication check of software according to documentation and through functional tests.</td><td><b>Validation Guidance</b> (in addition to the guidance for risk classes B and C)<b>:</b>
<i>Checks based on documentation:</i>
• Check whether the measures taken are appropriate with respect to the required state of the art for a high protection level.</td></tr>
<tr><td colspan="3"><b>Acceptable solutions:</b>
1. <b>Authenticity</b> For integrity reasons (see E3.3) an electronic signature is generated over the software part to be downloaded. Authenticity is guaranteed if a key stored in the fixed software part of the instrument confirms that the signature originates from the manufacturer. Key matching shall be done automatically.
2. <b>NB.</b> The key is stored in the fixed software part before initial verification.
3. <b>Correct type of measuring instrument</b>
Checking the instrument type requires automatically matching an identification of instrument type that is stored in the fixed software part of the instrument with a compatibility list attached to the software.</td></tr>
<tr><td colspan="2">4. <b>NB Approval</b>
If authenticity is guaranteed through the use of the manufacturer’s key, then NB approval may be assumed.</td><td>4. <b>NB Approval</b>
To check that software has been genuinely approved, one solution is that all downloaded approved software contains the responsible authority’s signature. The responsible authority’s public key is stored in the measuring instrument and is used to automatically check the signature attached to the software. It can be visualised at the instrument for comparison with the key published by the responsible authority.</td></tr>
</table>

Additions for Risk Class E
Required Documentation (in addition to the documentation required for risk classes B and C)**:** Source code of the fixed software part responsible for checking the authenticity of the downloaded software.
Validation Guidance (in addition to the guidance for risk classes B and C)**:** *Checks based on the source code:* • Check whether measures taken for checking the authenticity are appropriate.

<table>
<tr><th>Risk Class B</th><th>Risk Class C</th><th>Risk Class D</th></tr>
<tr><td colspan="3"><b>D3: Integrity of downloaded software</b>
<i>Means shall be employed to guarantee that the downloaded software has not been inadmissibly changed during download.</i></td></tr>
<tr><td colspan="3"><b>Specifying Notes:</b>
1. Before the downloaded software is used for the first time, the measuring instrument shall automatically check that the downloaded software has not been inadmissibly changed.
2. If the downloaded software fails this test, see D1.</td></tr>
<tr><td colspan="2"><b>Required Documentation:</b>
The documentation shall describe how the integrity of the software is guaranteed.</td><td><b>Required Documentation</b> (in addition to the documentation required for risk classes B and C)<b>:</b>
The measures of ensuring integrity shall be shown by the documentation.</td></tr>
<tr><td colspan="2"><b>Validation Guidance:</b>
• Ensure the integrity check of software after downloading according to documentation and through functional tests.</td><td><b>Validation Guidance</b> (in addition to the guidance for risk classes B and C)<b>:</b>
<i>Checks based on documentation:</i>
• Check whether the measures taken are appropriate with respect to the required state of the art for a high protection level.</td></tr>
<tr><td colspan="2"><b>Acceptable Solutions:</b>
• Integrity may be demonstrated by performing a checksum over the legally relevant software and comparing it against the checksum attached to the software (see also U2 for acceptable solutions).
• Acceptable algorithm: CRC, secret initial vector, length 32 bit. The initial vector is stored in the fixed software part.</td><td><b>Acceptable Solutions:</b>
• Generate a hash value of the software to be downloaded (algorithms e.g. SHA-1, RipeMD-160) and encrypt it (RSA, Elliptic Curves) with an appropriate key length.
• The key for decrypting is stored in the fixed software part.</td></tr>
</table>

Additions for Risk Class E
Required Documentation (in addition to the documentation required for risk classes B and C)**:** Source code of the fixed software part responsible for checking the integrity of the downloaded software.
Validation Guidance (in addition to the guidance for risk classes B and C)**:** *Checks based on the source code:* • Check whether measures taken for checking the integrity are appropriate.

Risk Class B	Risk Class C	Risk Class D
D4: Traceability of legally relevant software download *It shall be guaranteed by appropriate technical means that downloads of legally relevant software are adequately traceable within the instrument for subsequent controls.*		
Specifying Notes: 1. This requirement enables inspection authorities, which are responsible for the metrological surveillance of legally controlled instruments, to back-trace downloads of legally relevant software over an adequate period of time (that depends on national legislation). 2. The traceability means and records are part of the legally relevant software and should be protected as such.		
Required Documentation: The documentation shall: • Briefly describe how the traceability means is implemented and protected. • State how downloaded software may be traced.		**Required Documentation** (in addition to the documentation required for risk classes B and C)**:** The measures of ensuring traceability shall be shown by the documentation.
Validation Guidance: *Checks based on documentation:* • Check that traceability means are implemented and protected. *Functional checks:* • Check the functionality of the means through spot checks.		**Validation Guidance** (in addition to the guidance for risk classes B and C)**:** *Checks based on documentation:* • Check whether the measures taken are appropriate with respect to the required state of the art for a high protection level.
Acceptable Solutions: • An audit trail. The measuring instrument may be equipped with an event logger that automatically records at least the date and time of the download, identification of the downloaded legally relevant software, the identification of the downloading party, and an entry of the success. An entry is generated for each download attempt regardless of the success. • After having reached the limit of the event logger, it shall be ensured by technical means that further downloads are impossible. Audit trails may only be erased by breaking a physical or electronic seal and may be resealed only by the inspection authorities.		

Additions for Risk Class E
Required Documentation (in addition to the documentation required for risk classes B and C)**:** Source code of the fixed software part responsible for tracing download processes and managing the audit trail.
Validation Guidance (in addition to the guidance for risk classes B and C)**:** *Checks based on the source code:* • Check whether measures taken for tracing the download process are appropriate. • Check whether measures taken for protecting the audit trail are appropriate.

Risk Class B	Risk Class C	Risk Class D

D5: Download consent

It shall be guaranteed by technical means that software may only be loaded with the explicit consent of the user or owner of the measuring instrument, as appropriate.

Specifying Notes:

1. Once an instrument has been put into service the user, or its owner, is responsible for it. This requirement ensures that the manufacturer cannot change the legally relevant software of the instrument without the explicit consent of the appropriate responsible party.
2. The means by which the user / owner expresses his consent are part of the legally relevant software and should be protected as such. His consent is required by default unless he agrees otherwise.

~~3.The release by the user / owner applies for one download. It is reset after the download process or after a timeout.~~

~~4.~~3. The readiness of the device for download shall be indicated to the user / owner.

Required Documentation:

The documentation shall briefly describe the technical means by which the download process accounts for the consent of the user / owner .

Validation Guidance:

Checks based on documentation:

- Check by documentation which technical means are implemented to protect from download of legally relevant software without the explicit consent of the user.

Functional checks:

- Check that after each download a new consent from the user is required for a new download.
- Check through spot checks that a software download without user consent is prevented from.

Acceptable Solutions:

- A switch or key are acceptable when the user/owner is present to initiate download.
- For remote download, the measuring instrument's legally relevant software could contain a secure software switch that the user/owner could set to permit remote downloads in his absence.
- The consent could~~should~~ be limited to one download (or a specified number of downloads~~download~~), and there could ~~should~~ be a time-out after the permission has been given.
- If digital signatures are used to authenticate the sender, the latter's public key should be stored in the fixed software part of the measuring instrument. Automatic means would verify the authenticity of the signature attached to the software.

Additions for Risk Class E

Required Documentation (in addition to the documentation required for risk classes B, C and D)**:**

Source code of the fixed software part responsible for collecting the consent of the user / owner for a download.

Validation Guidance (in addition to the guidance for risk classes B, C and D)**:**

Checks based on the source code:

- Check whether measures taken for collecting the consent of the user / owner for a download are appropriate.

10 Extension I: Instrument Specific Software Requirements

This extension is intended to complement the general software requirements of the previous chapters and cannot be considered isolated from parts P or U and the other extensions (see chapter 3). It reflects the existence of instrument-specific MID annexes MI-x and contains specific aspects and requirements for measuring instruments or systems (or sub-assemblies). These requirements do not, however, go beyond the requirements of the MID. If reference is made to OIML recommendations or ISO/IEC standards this is done only if these can be considered as normative documents in the sense of the MID and if this supports a harmonised interpretation of the MID requirements.

Besides instrument specific software aspects and requirements Extension I contains the instrument (or category) specific assignment of risk classes which ensures a harmonised level of software examination, software protection and software conformity.

For the present, Extension I is intended to be an initial draft to be completed by the respective WELMEC Working Group that has the corresponding specific knowledge. Therefore Extension I has an "open structure", ie. it provides a skeleton that is - besides the initial assignment of risk classes - filled-in only partly (eg. for utility meters and automatic weighing instruments). It may be used for other MID (or non-MID instruments), too, according to the experiences gained and decisions taken by the responsible WELMEC Working Groups. The numbering x of the sub-chapters 10.x follows the numbering of the specific MID Annex MI-x. Non-MID instruments could be added starting from 10.11.

There are different instrument specific software aspects that might need consideration for a certain type x of measuring instrument. These aspects should be treated in a systematic manner as follows: Each sub-chapter 10.x should be subdivided into sections 10.x.y where y covers the following aspects.

10.x.1 Specific regulations, standards and other normative documents

Here, instrument (or category) specific regulations, standards and other normative documents (eg. OIML recommendations) or WELMEC guidelines should be mentioned that may help to develop instrument (or category) specific software requirements as an interpretation of the requirements of the MID Annex I and the specific annexes MI-x

Normally the specific software requirements apply in addition to the general ones in the previous chapters. Otherwise it should be clearly stated whether a specific software requirement replaces one (or more) of the general software requirements, or whether one (or more) general software requirements is (are) not applicable, and the reason why.

10.x.2 Technical description

Here
- examples of most common specific technical configurations,
- the application of parts P, U and extensions to these examples, and
- useful (instrument specific) checklists for both the manufacturer and the examiner

may be given.

The description should mention

- the measuring principle (cumulative measurement or single independent measurement; repeatable or non-repeatable measurement; static or dynamic measurement), and
- the fault detection and reaction;
  two cases are possible:
  a) the presence of a defect is obvious or can simply be checked or there are hardware means for fault detection,
  b) the presence of a defect is not obvious and cannot be easily checked and there are no hardware means for fault detection.

  In the latter case (b) fault detection and reaction requires appropriate software means and hence appropriate software requirements.
- the hardware configuration;
  at least the following issues should be addressed:
  - Is there a modular, general-purpose computer-based system or a dedicated instrument with an embedded system subject to legal control? Does the computer system stand-alone, or is it part of a closed network, e.g. Ethernet, token-ring LAN, or part of an open network, e.g. Internet?
  - Is the sensor separated (separate housing and separate power supply) from the Type U system or is it partly or completely integrated into it?
  - Is the user interface always under legal control (both for Type P and Type U instruments) or can it be switched to an operating mode which is not under legal control?
  - Is long-term data storage foreseen? If yes, then is the storage local (eg. hard disk) or remote (eg. file server)? Is the storage medium fixed (eg. internal ROM) or removable (eg. floppy disc, CD-RW, smart-media card, memory stick)?

the software configuration and environment;
at least the following issues should be addressed:
- Which operating system is used or can be used?
- Do other software applications reside on the system besides the legally relevant software?
- Is there software not subject to legal control that is intended to be freely modified after approval?

10.x.3 Specific software requirements

Here, the specific software requirements should be listed and commented using a similar form as in the previous chapters.

10.x.4 Examples of legally relevant functions and data

Here, examples of
- device specific parameters (eg. individual configuration and calibration parameters of a specific measuring instrument),
- type specific parameters (eg. specific parameters that are fixed at type approval), or
- legally relevant, specific functions

may be given.

10.x.5 Other aspects

Here, other aspects, eg. specific documentation required for type (software) examination, specific descriptions, and instructions to be supplied in type approval certificates, or other aspects (eg. requirements concerning the testability) may be mentioned.

10.x.6 Assignment of risk class

Here, the appropriate risk class for instruments of type x should be defined. This can be done

- either generally (for all categories within the respective type), or
- depending on the field of application, or category, or other aspects if these exist.

10.1 Water Meters

10.1.1 Specific regulations, standards and other normative documents

Water meters in residential, commercial and light industrial use are subject to regulations in MID. The specific requirements of this chapter are based on Annex MI-001 only.

The OIML recommendations R49and R72 are normative documents in the sense of the MID, but they have not yet been taken into consideration.

10.1.2 Technical description

10.1.2.1 Hardware Configuration

Water meters typically are realised as built-for purpose devices (Type P in this document). They may have one or more inputs for external sensor units.

10.1.2.2 Software Configuration

This is specific to each manufacturer but would normally expect to follow the recommendations given in the main body of this guide.

10.1.2.3 Measuring Principle

Water meters are continuously cumulating the volume consumed. They count impulses that are weighted with a certain incremental volume and calculate the total volume. The cumulated volume is displayed at the instrument.

The volume measurement is not repeatable.

10.1.2.4 Defects

The requirement MI-001, 7.1.2 deals with electromagnetic disturbances. There is need to interprete this requirement for software controlled instruments because detection of a disturbance and recovery is only possible by co-operation of specific hardware parts and specific software. From the software point of view it makes on the other hand no difference what the reason of a disturbance was (electromagnetic, electrical, mechanical etc): the recovery procedures are all the same.

10.1.3 Specific software requirements (Water meters)

<table>
<tr><th>Risk Class B</th><th>Risk Class C</th><th>Risk Class D</th></tr>
<tr><td colspan="3"><b>I1-1: Fault Detection</b>
<i>The software shall detect that normal processing is disturbed.</i></td></tr>
<tr><td colspan="3"><b>Specifying Notes:</b>
1. The fault detection software shall check that the appropriate metrological subroutines have been processed during the previous interval.
2. On detection of a fault:
a. The cumulative measurement and other relevant legal data shall be automatically saved to non-volatile storage (see Requirement I1-2).
b. The instrument shall try to restart (see requirement I1-3)</td></tr>
<tr><td colspan="3"><b>Required Documentation:</b>
A brief description of what is checked, what is required to trigger the fault detection process, what action is taken on the detection of a fault.</td></tr>
<tr><td colspan="3"><b>Validation Guidance:</b>
<i>Checks based on documentation:</i>
• Check whether the realisation of fault detection is appropriate.
<i>Functional checks:</i>
• If possible: simulate certain hardware faults and check whether they are detected by the software.</td></tr>
<tr><td colspan="3"><b>Acceptable solution:</b>
A hardware watchdog is reset by a cyclically processed microprocessor subroutine in order to inhibit the firing of the watchdog. Before resetting, the subroutine checks the health of the system eg. whether all metrologically relevant subroutines have been processed during the last interval. If any function has not been processed or - in the worst case - the microprocessor hang in an arbitrary endless loop, the reset of the watchdog doesn’t happen and it fires after a certain time span.</td></tr>
</table>

<table>
<tr><th>Risk Class B</th><th>Risk Class C</th><th>Risk Class D</th></tr>
<tr><td colspan="3"><b>I1-2: Back-up Facilities</b>
<i>There shall be a fall-back facility that allows the back-up of essential data such as. measurement values and the current status of the process in case of a disturbance. The state characteristics and important data shall be stored in a non-volatile storage.</i></td></tr>
<tr><td colspan="3"><b>Specifying Notes:</b>
Periodic backing up is acceptable if a controlled storage facility is not available due to hardware or functional constraints. However, the storage intervals must be small so that the discrepancy between the current and saved cumulative values is small compared to the maximum permissible error.</td></tr>
<tr><td colspan="3"><b>Required Documentation:</b>
A brief description of which data is backed up and when this occurs. Calculation of the maximum error that can occur for cumulative values if a cyclical back-up is realised.</td></tr>
<tr><td colspan="3"><b>Validation Guidance:</b>
<i>Checks based on documentation:</i>
• Check whether all legally relevant data are saved in case of a disturbance.
<i>Functional checks:</i>
• Check by simulating a disturbance whether back-up mechanism works.</td></tr>
<tr><td colspan="3"><b>Acceptable solution:</b>
A hardware watchdog fires when it is not cyclically reset. This alarm actuates an interrupt in the microprocessor. The assigned interrupt routine collects measurement values, state values and other relevant data and stores them in a non-volatile storage eg. an EEPROM.
<i>Note:</i> It is assumed that the watchdog interrupt has highest interrupt priority and can dominate any normal processing or any arbitrary endless loop, ie. the program control always jumps to the interrupt routine if the watchdog fires.</td></tr>
</table>

<table>
<tr><th>Risk Class B</th><th>Risk Class C</th><th>Risk Class D</th></tr>
<tr><td colspan="3"><b>I1-3: Wake-up Facilities and Restoring</b>
<i>The software shall not get into an indefinite state by the disturbance. If the program processing is disturbed, the instrument shall continuously be triggered to get back into normal operating mode. If triggering succeeds, the software shall boot up in a controlled way, retrieving the previously stored state and measurement values.</i></td></tr>
<tr><td colspan="3"><b>Specifying Notes:</b>
1. This facility should ensure that the last valid measurement and other legal data are stored first (see I1-2).</td></tr>
<tr><td colspan="3"><b>Required Documentation:</b>
The documentation shall brief describe how this facility works.</td></tr>
<tr><td colspan="3"><b>Validation Guidance:</b>
<i>Checks based on documentation:</i>
• Check whether an appropriate wake-up procedure has been realised.
<i>Functional checks:</i>
• Check by simulating a disturbance whether wake-up mechanism works.</td></tr>
<tr><td colspan="3"><b>Acceptable solution:</b>
A hardware watchdog fires cyclically actuating the reset signal in the microprocessor (this is not the same signal as the watchdog alarm in “acceptable solution” I1-2). The assigned reset routine starts initialisation of the hardware and software data domains. Subsequently it retrieves the last measured values, the state values and other relevant data from the non-volatile storage.
<i>Note:</i> It is assumed that the watchdog reset signal has highest priority (even higher than that of the alarm interrupt in “acceptable solution” I1-2) and can dominate normal processing, ie. jump from any point in the program to the reset routine.</td></tr>
</table>

<table>
<tr><th>Risk Class B</th><th>Risk Class C</th><th>Risk Class D</th></tr>
<tr><td colspan="3"><b>I1-4: Internal Resolution</b>
<i>For guaranteeing a sufficient resolution of the indication the internal representation of the measurement value (the type of the variables) shall have a sufficient number of digits to ensure that the quantity passed does not return the digits to their initial values.</i></td></tr>
<tr><td colspan="3"><b>Specifying Notes:</b></td></tr>
<tr><td colspan="3"><b>Required Documentation:</b>
Documentation of the internal representation of the volume register and auxiliary quantities (variable types).</td></tr>
<tr><td colspan="3"><b>Validation Guidance:</b>
<i>Checks based on documentation:</i>
• Check whether internal resolution is sufficient.</td></tr>
<tr><td colspan="3"><b>Acceptable solution:</b>
For the internal software representation at least two more digits for the decimal fractal are necessary, than for the indication. The variable type in the software is chosen accordingly.</td></tr>
</table>

<table>
<tr><th>Risk Class B</th><th>Risk Class C</th><th>Risk Class D</th></tr>
<tr><td colspan="3"><b>I1-5: MID-Annex I, 8.5</b> (Inhibit resetting of cumulative measurement values)
<i>For utility measuring instruments the display of the total quantity supplied or the displays from which the total quantity supplied can be derived, whole or partial reference to which is the basis for payment, shall not be able to be reset during use.</i></td></tr>
<tr><td colspan="3"><b>Specifying Notes:</b>
Cumulative registers of an measuring instrument may be reset prior to being put into use.</td></tr>
<tr><td colspan="3"><b>Required Documentation:</b>
Documentation of protection means against resetting the volume registers.</td></tr>
<tr><td colspan="3"><b>Validation Guidance:</b>
<i>Checks based on documentation:</i>
• Check whether cumulative legally relevant measurement values cannot be reset without leaving a trace.
<i>Functional checks:</i>
• Check whether the respective values cannot be reset.</td></tr>
<tr><td colspan="3"><b>Acceptable solution:</b>
The registers for volume are protected against changes and resetting by the same means as parameters (see P7).</td></tr>
</table>

Risk Class B	Risk Class C	Risk Class D
I1-6: MID-Annex I, 10.5 (Indication for the customer) *Whether or not a measuring instrument intended for utility measurement purposes can be remotely read it shall in any case be fitted with a metrologically controlled display accessible without tools to the consumer. The reading of this display is the measurement result that serves as the basis for the price to pay.*		
Specifying Notes:		
Required Documentation: Documentation of the hardware of the instrument.		
Validation Guidance: *Checks based on documentation:* • Check whether the hardware configuration contains a display for the measuring values.		
Acceptable solution: One solution is integrating the display in the housing of the instrument. Another is the connection of a hardware unit with indication. The hardware unit and the transmission line or network have to fulfil the requirements of P or U and Extension T.		

10.1.4 Examples of legally relevant functions and data

Water meters are characterised by several typical parameters that are used as constants for calculations, as configuration parameters etc but also for setting up the functionality of the device. Concerning identification and protection of parameters and parameter sets refer to requirement P2 and P7, guide P. In the following some examples of typical parameters of water meters are given.

Parameter	Protected	Settable	Comment
Calibration factor	x		Impulses per m^3
Linearisation factor	x		
Interface parameters		x	Baud-rate etc

10.1.5 Other aspects

None

10.1.6 Assignment of risk class

For the present, according to the result of the WELMEC WG7 questionnaire (2004) and subject to future considerations of the responsible WELMEC Working Group, the following risk class should be applied if software examinations based on this guide are carried out for (software-controlled) water meters:

- risk class B for instruments of type P

For instruments of type U there are no experiences yet, but a higher risk class (eg. C) seems appropriate.

10.2 Gas Meters and Volume Conversion Devices

10.2.1 Specific regulations, standards and other normative documents

Gas meters and volume conversion devices in residential, commercial and light industrial use are subject to regulations in MID. The specific requirements of this chapter are based on Annex MI-002 only.

The OIML recommendations R6, R31and R32 are normative documents in the sense of the MID, but they have not yet been taken into consideration.

10.2.2 Technical description

10.2.2.1 Hardware Configuration

Gas meters and volume conversion devices typically are realised as built-for purpose devices (Type P in this document). They may have one or more inputs for external sensor units and meter and conversion device may be different hardware units.~~ {... to be completed}~~

10.2.2.2 Software Configuration

This is specific to each manufacturer but would normally expect to follow the recommendations given in the main body of this guide.

10.2.2.3 Measuring Principle

Gas meters are continuously cumulating the volume consumed. They count impulses that are weighted with a certain incremental volume and calculate the total volume. The cumulated volume is displayed at the instrument. (Other principles ???) The volume may be converted to a temperature and pressure compensated value by a converter. ~~(???) {... to be completed}~~

The volume measurement is not repeatable.

10.2.2.4 Defects

The requirement MI-002, 3.1.2 deals with electromagnetic disturbances. There is a need to interprete this requirement for software controlled instruments because detection of a disturbance and recovery is only possible by co-operation of specific hardware parts and specific software. From the software point of view it makes on the other hand no difference what the reason of a disturbance was (electromagnetic, electrical, mechanical etc): the recovery procedures are all the same.

10.2.3 Specific software requirements (Gas meters and volume converters)

Risk Class B	Risk Class C	Risk Class D
I2-1: Fault Detection *The software shall detect that normal processing is disturbed.*		
Specifying Notes: 1. The fault detection software shall check that the appropriate metrological subroutines have been processed during the previous interval. 2. On detection of a fault: a. The cumulative measurement and other relevant legal data shall be automatically saved to non-volatile storage (see Requirement I2-2). b. The instrument shall try to restart (see requirement I2-3)		
Required Documentation: A brief description of what is checked, what is required to trigger the fault detection process, what action is taken on the detection of a fault.		
Validation Guidance: *Checks based on documentation:* • Check whether the realisation of fault detection is appropriate. *Functional checks:* • If possible: simulate certain hardware faults and check whether they are detected by the software.		
Acceptable solution: A hardware watchdog is reset by a cyclically processed microprocessor subroutine in order to inhibit the firing of the watchdog. Before resetting, the subroutine checks the health of the system eg. whether all metrologically relevant subroutines have been processed during the last interval. If any function has not been processed or - in the worst case - the microprocessor hang in an arbitrary endless loop, the reset of the watchdog doesn't happen and it fires after a certain time span.		

<table>
<tr><th>Risk Class B</th><th>Risk Class C</th><th>Risk Class D</th></tr>
<tr><td colspan="3"><b>I2-2: Back-up Facilities</b>
<i>There shall be a fall-back facility that allows to back-up essential data like eg. measurement values and the current status of the process in case of a disturbance. The state characteristics and important data shall be stored in a non-volatile storage.</i></td></tr>
<tr><td colspan="3"><b>Specifying Notes:</b>
Periodic backing up is acceptable if a controlled storage facility is not available due to hardware or functional constraints. However, the storage intervals must be small so that the discrepancy between the current and saved cumulative values is small compared to the maximum permissible error.</td></tr>
<tr><td colspan="3"><b>Required Documentation:</b>
A brief description of which data is backed up and when this occurs. Calculation of the maximum error that can occur for cumulative values if a cyclical back-up is realised.</td></tr>
<tr><td colspan="3"><b>Validation Guidance:</b>
<i>Checks based on documentation:</i>
• Check whether all legally relevant data are saved in case of a disturbance.
<i>Functional checks:</i>
• Check by simulating a disturbance whether back-up mechanism works.</td></tr>
<tr><td colspan="3"><b>Acceptable solution:</b>
A hardware watchdog fires when it is not cyclically reset. This alarm actuates an interrupt in the microprocessor. The assigned interrupt routine collects measurement values, state values and other relevant data and stores them in a non-volatile storage eg. an EEPROM.
<i>Note:</i> It is assumed that the watchdog interrupt has highest interrupt priority and can dominate any normal processing or any arbitrary endless loop, ie. the program control always jumps to the interrupt routine if the watchdog fires.</td></tr>
</table>

<table>
<tr><th>Risk Class B</th><th>Risk Class C</th><th>Risk Class D</th></tr>
<tr><td colspan="3"><b>I2-3: Wake-up Facilities and Restoring</b>
<i>The software shall not get into an indefinite state by the disturbance. If the program processing is disturbed, the instrument shall continuously be triggered to get back into normal operating mode. If triggering succeeds, the software shall boot up in a controlled way, retrieving the previously stored state and measurement values.</i></td></tr>
<tr><td colspan="3"><b>Specifying Notes:</b>
1. This facility should ensure that the last valid measurement and other legal data are stored first (see I2-2).</td></tr>
<tr><td colspan="3"><b>Required Documentation:</b>
The documentation shall brief describe how this facility works.</td></tr>
<tr><td colspan="3"><b>Validation Guidance:</b>
<i>Checks based on documentation:</i>
• Check whether an appropriate wake-up procedure has been realised.
<i>Functional checks:</i>
• Check by simulating a disturbance whether wake-up mechanism works.</td></tr>
<tr><td colspan="3"><b>Acceptable solution:</b>
A hardware watchdog fires cyclically actuating the reset signal in the microprocessor (this is not the same signal as the watchdog alarm in “acceptable solution” I2-2). The assigned reset routine starts initialisation of the hardware and software data domains. Subsequently it retrieves the last measured values, the state values and other relevant data from the non-volatile storage.
<i>Note:</i> It is assumed that the watchdog reset signal has highest priority (even higher than that of the alarm interrupt in “acceptable solution” I2-2) and can dominate normal processing, ie. jump from any point in the program to the reset routine.</td></tr>
</table>

Risk Class B	Risk Class C	Risk Class D
I2-4: MI-002, 5.3 (Internal Resolution) *For guaranteeing a sufficient resolution of the indication the internal representation of the measurement value (the type of the variables) shall have a sufficient number of digits to ensure that the quantity passed during 8000 hours at Q_{max} does not return the digits to their initial values.*		
Specifying Notes:		
Required Documentation: Documentation of the internal representation of the electrical energy register and auxiliary quantities (variable types).		
Validation Guidance: *Checks based on documentation:* • Check whether internal resolution is sufficient.		
Acceptable solution: For the internal software representation at least two more digits for the decimal fractal are necessary, than for the indication. The variable type in the software is chosen accordingly. Typical values for gas meters are: {... to be completed}		

Risk Class B	Risk Class C	Risk Class D
I2-5: MID-Annex I, 8.5 (Inhibit resetting of cumulative measurement values) *For utility measuring instruments the display of the total quantity supplied or the displays from which the total quantity supplied can be derived, whole or partial reference to which is the basis for payment, shall not be able to be reset during use.*		
Specifying Notes: Cumulative registers of an measuring instrument may be reset prior to being put into use.		
Required Documentation: Documentation of protection means against resetting the volume registers.		
Validation Guidance: *Checks based on documentation:* • Check whether cumulative legally relevant measurement values cannot be reset without leaving a trace. *Functional checks:* • Check whether the respective values cannot be reset.		
Acceptable solution: The registers for volume are protected against changes and resetting by the same means as parameters (see P7).		

Risk Class B	Risk Class C	Risk Class D
I2-6: MID-Annex I, 10.5 (Indication for the customer) *Whether or not a measuring instrument intended for utility measurement purposes can be remotely read it shall in any case be fitted with a metrologically controlled display accessible without tools to the consumer. The reading of this display is the measurement result that serves as the basis for the price to pay.*		
Specifying Notes:		
Required Documentation: Documentation of the hardware of the instrument.		
Validation Guidance: *Checks based on documentation:* • Check whether the hardware configuration contains a display for the measuring values.		
Acceptable solution: One solution is integrating the display in the housing of the instrument. Another is the connection of a hardware unit with indication. The hardware unit and the transmission line or network have to fulfil the requirements of P or U and Extension T.		

Risk Class B	Risk Class C	Risk Class D
I2-7: MI-002, 5.2 (Power source lifetime) *A dedicated power source shall have a lifetime of at least five years. After 90% of its lifetime an appropriate warning shall be shown.*		
Specifying Notes: Lifetime is used here in the sense of available energy capacity		
Required Documentation: Documentation of the battery capacity, battery maximum lifetime (independent of energy consumption), measures to determine the consumed or available energy, description of the means for the warning of low battery capacitance.		
Validation Guidance: *Checks based on documentation:* • Check whether the measures taken are appropriate for the surveillance of the battery status.		
Acceptable solution: The operating hours or the wake-up events of the device are counted, stored in a non-volatile memory and compared with the nominal value of the battery lifetime. If 90% of the lifetime has elapsed an appropriate warning is shown. The software detects the exchange of the battery and resets the counter. Another solution would be to monitor the health of the power supply continuously.		

Risk Class B	Risk Class C	Risk Class D
I2-8: MI-002, 9.1 (Electronic conversion device) *An electronic conversion device shall be capable of detecting when it is operating outside the operating range(s) stated by the manufacturer for parameters that are relevant for measurement accuracy. In such a case, the conversion device must stop integrating the converted quantity, and may totalise separately the converted quantity for the time it is operating outside the operating range(s).*		
Specifying Notes:		
Required Documentation: Documentation of the different registers for converted quantity and failure quantity.		
Validation Guidance: *Checks based on documentation:* • Check whether the measures taken are appropriate for the management of unusual operating conditions.		
Acceptable solution: The software monitors the relevant input values and compares them with predefined limits. If all values are inside the limits the converted quantity is integrated to the normal register (a dedicated variable). Else it totalises the quantity in another variable. Another solution would be to have only one cumulating register but to record the start and end date, time and register values of the out-of-range period in an event logger (see P7). Both quantities can be indicated. The user can clearly identify and distinguish the regular and the failure indication by means of a status indication.		

Risk Class B	Risk Class C	Risk Class D
I2-9: MI-002, 5.5 (Test element) *The gas meter shall have a test element, which shall enable tests to be carried out in a reasonable time.*		
Specifying Notes: The test element for accelerating lengthy procedures is normally used for testing before installation and normal operation.		
Required Documentation: Documentation of the test element and instructions for activating the test mode.		
Validation Guidance: *Checks based on documentation:* • Check whether all lengthy processes of the gas meter can be tested by means of the test element.		
Acceptable solution: The time base of the internal clock can be accelerated. Processes that last eg. a week, a month or even a year and overrun of registers may be tested in the test mode within a time span of minutes or hours.		

10.2.4 Examples of legally relevant functions and data

Gas meters and volume converters often have a lot of parameters. They are used as constants for calculations, as configuration parameters etc, but also for setting up the functionality of the device. Concerning identification and protection of parameters and parameter sets refer to requirement P2 and P7, guide P.

In the following some examples of typical parameters of utility meters are given.

Parameter	Protected	Settable	Comment
Calibration factor	x		Impulses per m^3
Linearisation factor	x		
Interface parameters		x	Baud-rate etc

10.2.5 Other aspects

None

10.2.6 Assignment of risk class

For the present, according to the result of the WELMEC WG7 questionnaire (2004) and subject to future considerations of the responsible WELMEC Working Group, the following risk class should be applied if software examinations based on this guide are carried out for (software-controlled) gas meters and volume conversion devices:

- risk class B for instruments and devices of type P

- risk class C for instruments and devices of type U

10.3 Active Electrical Energy Meters

10.3.1 Specific regulations, standards and other normative documents

Active electrical energy meters in residential, commercial and light industrial use are subject to regulations in MID. The specific requirements of this chapter are based on Annex MI-003 only.

The OIML recommendation R46 and the standard ISO/IEC 61036 are normative documents in the sense of the MID, but they have not yet been taken into consideration.

10.3.2 Technical description

Active electrical energy meters measure voltage and current, calculate the active electrical power, and integrate this to the active electrical energy.

Note: There are electronic meters with software on the market which do not use a digital measuring principle. They could be based on Hall sensors, logarithmic amplifiers etc. Most of the micro-processor or ASIC-based meters use a principle where the instantaneous values of the (AC) voltage and current are multiplied, and the product integrated. This gives pr. definition the active electrical energy. No measurement of the power factor is done. In principle it is possible to measure the power by measuring the RMS values of current and voltage plus the power factor, and multiply these three quantities.

10.3.2.1 Hardware Configuration

Active electrical energy meters typically are realised as built-for purpose devices (Type P in this document). They may have one or more inputs for external sensor units and may be used in combination with external instrument transformers.

10.3.2.2 Software Configuration

This is specific to each manufacturer but would normally expect to follow the recommendations given in the main body of this guide.

10.3.2.3 Measuring Principle

Active electrical energy meters are continuously cumulating the energy consumed in a circuit. They measure true instantaneous AC voltage and current and calculate the electrical energy from these quantities, taking the power factor cos φ into account. The cumulative energy value is displayed at the instrument.

The energy measurement is not repeatable.

10.3.2.4 Defects

The requirement MI-003, 4.3.1 deals with electromagnetic disturbances. There is a need to interprete this requirement for software controlled instruments because detection of a disturbance and recovery is only possible by co-operation of specific hardware parts and specific software. From the software point of view it makes on the other hand no difference what the reason of a disturbance was (electromagnetic, electrical, mechanical etc): the recovery procedures are all the same.

10.3.3 Specific software requirements (Active electrical energy meters)

Risk Class B	Risk Class C	Risk Class D

I3-1: Fault Detection
The software shall detect that normal processing is disturbed.

Specifying Notes:

1. The fault detection software shall check that the appropriate metrological subroutines have been processed during the previous interval.
2. On detection of a fault:
   a. The cumulative measurement and other relevant legal data shall be automatically saved to non-volatile storage (see Requirement I3-2).
   b. The instrument shall try to restart (see requirement I3-3)

Required Documentation:
A brief description of what is checked, what is required to trigger the fault detection process, what action is taken on the detection of a fault.

Validation Guidance:

Checks based on documentation:

- Check whether the realisation of fault detection is appropriate.

Functional checks:

- If possible: simulate certain hardware faults and check whether they are detected by the software.

Acceptable solution:

A hardware watchdog is reset by a cyclically processed microprocessor subroutine in order to inhibit the firing of the watchdog. Before resetting, the subroutine checks the health of the system eg. whether all metrologically relevant subroutines have been processed during the last interval. If any function has not been processed or - in the worst case - the microprocessor hang in an arbitrary endless loop, the reset of the watchdog doesn't happen and it fires after a certain time span.

Risk Class B	Risk Class C	Risk Class D

I3-2: Back-up Facilities

There shall be a fall-back facility that allows to back-up essential data like eg. measurement values and the current status of the process in case of a disturbance. The state characteristics and important data shall be stored in a non-volatile storage.

Specifying Notes:

Periodic backing up is acceptable if a controlled storage facility is not available due to hardware or functional constraints. However, the storage intervals must be small so that the discrepancy between the current and saved cumulative values is small compared to the maximum permissible error.

Required Documentation:

A brief description of which data is backed up and when this occurs. Calculation of the maximum error that can occur for cumulative values if a cyclical back-up is realised.

Validation Guidance:

Checks based on documentation:

- Check whether all legally relevant data are saved in case of a disturbance.

Functional checks:

- Check by simulating a disturbance whether back-up mechanism works.

Acceptable solution:

A hardware watchdog fires when it is not cyclically reset. This alarm actuates an interrupt in the microprocessor. The assigned interrupt routine collects measurement values, state values and other relevant data and stores them in a non-volatile storage eg. an EEPROM.

Note: It is assumed that the watchdog interrupt has highest interrupt priority and can dominate any normal processing or any arbitrary endless loop, ie. the program control always jumps to the interrupt routine if the watchdog fires.

Risk Class B	Risk Class C	Risk Class D

I3-3: Wake-up Facilities and Restoring

The software shall not get into an indefinite state by the disturbance. If the program processing is disturbed, the instrument shall continuously be triggered to get back into normal operating mode. If triggering succeeds, the software shall boot up in a controlled way, retrieving the previously stored state and measurement values.

Specifying Notes:

1. This facility should ensure that the last valid measurement and other legal data are stored first (see I3-2).

Required Documentation:

The documentation shall brief describe how this facility works.

Validation Guidance:

Checks based on documentation:

- Check whether an appropriate wake-up procedure has been realised.

Functional checks:

- Check by simulating a disturbance whether wake-up mechanism works.

Acceptable solution:

A hardware watchdog fires cyclically actuating the reset signal in the microprocessor (this is not the same signal as the watchdog alarm in “acceptable solution” I3-2). The assigned reset routine starts initialisation of the hardware and software data domains. Subsequently it retrieves the last measured values, the state values and other relevant data from the non-volatile storage.

Note: It is assumed that the watchdog reset signal has highest priority (even higher than that of the alarm interrupt in “acceptable solution” I3-2) and can dominate normal processing, ie. jump from any point in the program to the reset routine.

<table>
<tr><th>Risk Class B</th><th>Risk Class C</th><th>Risk Class D</th></tr>
<tr><td colspan="3"><b>I3-4: MI-003, 5.2</b> (Internal Resolution)
<i>For guaranteeing a sufficient resolution of the indication the internal representation of the measurement value (the type of the variables) shall have a sufficient number of digits to ensure that the quantity passed during 4000 hours at P_{max} does not return the digits to their initial values.</i></td></tr>
<tr><td colspan="3"><b>Specifying Notes:</b></td></tr>
<tr><td colspan="3"><b>Required Documentation:</b>
Documentation of the internal representation of the electrical energy register and auxiliary quantities (variable types).</td></tr>
<tr><td colspan="3"><b>Validation Guidance:</b>
<i>Checks based on documentation:</i>
• Check whether internal resolution is sufficient.</td></tr>
<tr><td colspan="3"><b>Acceptable solution:</b>
Typical values for electricity meters are: P_{max} = 3*60 A * 230 V. The required range is 165600 kWh. For the internal software representation at least two more digits for the decimal fractal are necessary, ie. the appropriate types of the register for energy would be 32 bit integer but IEEE 4 byte floating point (single precision, 7 significant decimal digits) may not be appropriate.</td></tr>
</table>

<table>
<tr><th>Risk Class B</th><th>Risk Class C</th><th>Risk Class D</th></tr>
<tr><td colspan="3"><b>I3-5: MID-Annex I, 8.5</b> (Inhibit resetting of cumulative measurement values)
<i>For utility measuring instruments the display of the total quantity supplied or the displays from which the total quantity supplied can be derived, whole or partial reference to which is the basis for payment, shall not be able to be reset during use.</i></td></tr>
<tr><td colspan="3"><b>Specifying Notes:</b>
Cumulative registers of an measuring instrument may be reset prior to being put into use.</td></tr>
<tr><td colspan="3"><b>Required Documentation:</b>
Documentation of protection means against resetting the energy registers.</td></tr>
<tr><td colspan="3"><b>Validation Guidance:</b>
<i>Checks based on documentation:</i>
• Check whether cumulative legally relevant measurement values cannot be reset without leaving a trace.
<i>Functional checks:</i>
• Check whether the respective values cannot be reset.</td></tr>
<tr><td colspan="3"><b>Acceptable solution:</b>
The registers for energy are protected against changes and resetting by the same means as parameters (see P7).</td></tr>
</table>

Risk Class B	Risk Class C	Risk Class D
I3-6: MID-Annex I, 10.5 (Indication for the customer) *Whether or not a measuring instrument intended for utility measurement purposes can be remotely read it shall in any case be fitted with a metrologically controlled display accessible without tools to the consumer. The reading of this display is the measurement result that serves as the basis for the price to pay.*		
Specifying Notes:		
Required Documentation: Documentation of the hardware of the instrument.		
Validation Guidance: *Checks based on documentation:* • Check whether the hardware configuration contains a display for the measuring values.		
Acceptable solution: One solution is integrating the display in the housing of the instrument. Another is the connection of a hardware unit with indication. The hardware unit and the transmission line or network have to fulfil the requirements of P or U and Extension T.		

10.3.4 Examples of legally relevant functions and data

Electronic utility meters often have a lot of parameters. They are used as constants for calculations, as configuration parameters etc but also for setting up the functionality of the device. Concerning identification and protection of parameters and parameter sets refer to requirement P2 and P7, guide P.
In the following some typical parameters of utility meters are given.

Parameter	Protected	Settable	Comment
Calibration factor	x		Impulses per kWh, instrument transformer ratio etc
Linearisation factor	x		
Interface parameters		x	Baud-rate etc

10.3.5 Other aspects

None

10.3.6 Assignment of risk class

For the present, according to the result of the WELMEC WG7 questionnaire (2004) and subject to future decisions of the responsible WELMEC Working Group, the following risk class should be applied if software examinations based on this guide are carried out for (software-controlled) active electrical energy meters:

- risk class B for instruments of type P (except prepayment and interval meters)
- risk class C for prepayment and interval meters of type P

For instruments of type U there are no experiences yet, but a higher risk class (C or even D, probably depending on the field of application) seems appropriate and should be discussed by the responsible WELMEC Working Group if required.

10.4 Heat Meters

10.4.1 Specific regulations, standards and other normative documents

Heat meters in residential, commercial and light industrial use are subject to regulations in MID. The specific requirements of this chapter are based on Annex MI-004.

At least the OIML recommendation R75 is a normative document in the sense of the MID, but it has not yet been taken into consideration.

10.4.2 Technical description

10.4.2.1 Hardware Configuration

Heat meters typically are realised as built-for purpose devices (Type P in this document). They consist of a differential temperature conversion device and a volume meter. The temperature conversion devices may be *sub-assemblies* according to MID Article 4. ~~{... to be completed}~~

10.4.2.2 Software Configuration

This is specific to each manufacturer but would normally expect to follow the recommendations given in the main body of this guide.

10.4.2.3 Measuring Principle

Heat meters are continuously cumulating the energy consumed in a heating circuit. ~~... (siehe 50.7)~~ The cumulated thermal energy value is displayed at the instrument.

The energy measurement is not repeatable.

10.4.2.4 Defects

The requirement MI-004, 4.1 deals with electromagnetic disturbances. There is a need to interprete this requirement for software controlled instruments because detection of a disturbance and recovery is only possible by co-operation of specific hardware parts and specific software. From the software point of view it makes on the other hand no difference what the reason of a disturbance was (electromagnetic, electrical, mechanical etc): the recovery procedures are all the same.

10.4.3 Specific software requirements (Heat meters)

Risk Class B	Risk Class C	Risk Class D

I4-1: Fault Detection
The software shall detect that normal processing is disturbed.

Specifying Notes:

1. The fault detection software shall check that the appropriate metrological subroutines have been processed during the previous interval.
2. On detection of a fault:
   a. The cumulative measurement and other relevant legal data shall be automatically saved to non-volatile storage (see Requirement I4-2).
   b. The instrument shall try to restart (see requirement I4-3)

Required Documentation:
A brief description of what is checked, what is required to trigger the fault detection process, what action is taken on the detection of a fault.

Validation Guidance:

Checks based on documentation:
- Check whether the realisation of fault detection is appropriate.

Functional checks:
- If possible: simulate certain hardware faults and check whether they are detected by the software.

Acceptable solution:

A hardware watchdog is reset by a cyclically processed microprocessor subroutine in order to inhibit the firing of the watchdog. Before resetting, the subroutine checks the health of the system eg. whether all metrologically relevant subroutines have been processed during the last interval. If any function has not been processed or - in the worst case - the microprocessor hang in an arbitrary endless loop, the reset of the watchdog doesn’t happen and it fires after a certain time span.

Risk Class B	Risk Class C	Risk Class D

I4-2: Back-up Facilities
There shall be a fall-back facility that allows to back-up essential data like eg. measurement values and the current status of the process in case of a disturbance. The state characteristics and important data shall be stored in a non-volatile storage.

Specifying Notes:

Periodic backing up is acceptable if a controlled storage facility is not available due to hardware or functional constraints. However, the storage intervals must be small so that the discrepancy between the current and saved cumulative values is small compared to the maximum permissible error.

Required Documentation:

A brief description of which data is backed up and when this occurs. Calculation of the maximum error that can occur for cumulative values if a cyclical back-up is realised.

Validation Guidance:

Checks based on documentation:
- Check whether all legally relevant data are saved in case of a disturbance.

Functional checks:
- Check by simulating a disturbance whether back-up mechanism works.

Acceptable solution:

A hardware watchdog fires when it is not cyclically reset. This alarm actuates an interrupt in the microprocessor. The assigned interrupt routine collects measurement values, state values and other relevant data and stores them in a non-volatile storage eg. an EEPROM.

Note: It is assumed that the watchdog interrupt has highest interrupt priority and can dominate any normal processing or any arbitrary endless loop, ie. the program control always jumps to the interrupt routine if the watchdog fires.

Risk Class B	Risk Class C	Risk Class D

I4-3: Wake-up Facilities and Restoring
The software shall not get into an indefinite state by the disturbance. If the program processing is disturbed, the instrument shall continuously be triggered to get back into normal operating mode. If triggering succeeds, the software shall boot up in a controlled way, retrieving the previously stored state and measurement values.

Specifying Notes:

1. This facility should ensure that the last valid measurement and other legal data are stored first (see I4-2).

Required Documentation:
The documentation shall brief describe how this facility works.

Validation Guidance:

Checks based on documentation:
- Check whether an appropriate wake-up procedure has been realised.

Functional checks:
- Check by simulating a disturbance whether wake-up mechanism works.

Acceptable solution:
A hardware watchdog fires cyclically actuating the reset signal in the microprocessor (this is not the same signal as the watchdog alarm in “acceptable solution” I4-2). The assigned reset routine starts initialisation of the hardware and software data domains. Subsequently it retrieves the last measured values, the state values and other relevant data from the non-volatile storage.

Note: It is assumed that the watchdog reset signal has highest priority (even higher than that of the alarm interrupt in “acceptable solution” I4-2) and can dominate normal processing, ie. jump from any point in the program to the reset routine.

<table>
<tr><th>Risk Class B</th><th>Risk Class C</th><th>Risk Class D</th></tr>
<tr><td colspan="3"><b>I4-4: Internal Resolution</b>
<i>For guaranteeing a sufficient resolution of the indication the internal representation of the measurement value (the type of the variables) shall have a sufficient number of digits to ensure that the quantity passed does not return the digits to their initial values.</i></td></tr>
<tr><td colspan="3"><b>Specifying Notes:</b></td></tr>
<tr><td colspan="3"><b>Required Documentation:</b>
Documentation of the internal representation of the thermal energy register and auxiliary quantities (variable types).</td></tr>
<tr><td colspan="3"><b>Validation Guidance:</b>
<i>Checks based on documentation:</i>
• Check whether internal resolution is sufficient.</td></tr>
<tr><td colspan="3"><b>Acceptable solution:</b>
For the internal software representation at least two more digits for the decimal fractal are necessary, than for the indication. The variable type in the software is chosen accordingly.</td></tr>
</table>

<table>
<tr><th>Risk Class B</th><th>Risk Class C</th><th>Risk Class D</th></tr>
<tr><td colspan="3"><b>I4-5: MID-Annex I, 8.5</b> (Inhibit resetting of cumulative measurement values)
<i>For utility measuring instruments the display of the total quantity supplied or the displays from which the total quantity supplied can be derived, whole or partial reference to which is the basis for payment, shall not be able to be reset during use.</i></td></tr>
<tr><td colspan="3"><b>Specifying Notes:</b>
Cumulative registers of an measuring instrument may be reset prior to being put into use.</td></tr>
<tr><td colspan="3"><b>Required Documentation:</b>
Documentation of protection means against resetting the energy registers.</td></tr>
<tr><td colspan="3"><b>Validation Guidance:</b>
<i>Checks based on documentation:</i>
• Check whether cumulative legally relevant measurement values cannot be reset without leaving a trace.
<i>Functional checks:</i>
• Check whether the respective values cannot be reset.</td></tr>
<tr><td colspan="3"><b>Acceptable solution:</b>
The registers for energy are protected against changes and resetting by the same means as parameters (see P7).</td></tr>
</table>

Risk Class B	Risk Class C	Risk Class D
I4-6: MID-Annex I, 10.5 (Indication for the customer) *Whether or not a measuring instrument intended for utility measurement purposes can be remotely read it shall in any case be fitted with a metrologically controlled display accessible without tools to the consumer. The reading of this display is the measurement result that serves as the basis for the price to pay.*		
Specifying Notes:		
Required Documentation: Documentation of the hardware of the instrument.		
Validation Guidance: *Checks based on documentation:* • Check whether the hardware configuration contains a display for the measuring values.		
Acceptable solution: One solution is integrating the display in the housing of the instrument. Another is the connection of a hardware unit with indication. The hardware unit and the transmission line or network have to fulfil the requirements of P or U and Extension T.		

10.4.4 Examples of legally relevant functions and data

Heat meters often have many parameters. They are used as constants for calculations, as configuration parameters etc but also for setting up the functionality of the device. Concerning identification and protection of parameters and parameter sets refer to requirement P2 and P7, guide P.
In the following, some examples of typical parameters of heat meters are given.

Parameter	Protected	Settable	Comment
Calibration factor	x		Impulses per m^3, compensation of different bias and sensibility of the temperatur sensor
Linearisation factor	x		
Interface parameters		x	Baud-rate etc

10.4.5 Other aspects

None

10.4.6 Assignment of risk class

For the present, according to the result of the WELMEC WG7 questionnaire (2004) and subject to future decisions of the responsible WELMEC Working Group, the following risk class should be applied if software examinations based on this guide are carried out for (software-controlled) heat meters:

- risk class B for instruments and sub-assemblies of type P
- risk class C for instruments and sub-assemblies of type U

10.5 Measuring Systems for the Continuous and Dynamic Measurement of Quantities of Liquids Other than Water

Measuring Systems for the Continuous and Dynamic Measurement of Quantities of Liquids Other than Water are subject to regulations in MID. The specific requirements are in Annex MI-005. Neither these specific requirements nor any normative documents have yet been taken into consideration.

10.5.1 - 10.5.5 will be filled in if considered necessary in the future.

10.5.6 Assignment of risk class

For the present, according to the result of the WELMEC WG7 questionnaire (2004) and subject to future decisions of the responsible WELMEC Working Group, the following risk class should be applied if software examinations based on this guide are carried out for (software-controlled) measuring systems for the continuous and dynamic measurement of quantities of liquids other than water.

- Risk class B for instruments and devices of type P if these belong to an uncritical category and an uncritical field of application
- Risk class C for instruments and devices of type P if these belong to a critical category or a critical field of application
- Risk class C for instruments and devices of type U if these belong to an uncritical category and an uncritical field of application
- Risk class D for instruments and devices of type U if these belong to a critical category or a critical field of application

The responsible WELMEC Working Group (WG10) should define the critical and uncritical categories and the critical and uncritical fields of application.

10.6 Weighing Instruments

Weighing instruments are divided into two main categories:

1. Non-automatic weighing instruments (NAWIs), and

2. Automatic weighing instruments (AWIs).

While most AWIs are governed by the MID, NAWIs are not; they are still governed by the European Directive 90/384/EEC. **Therefore the software guide WELMEC 2.3 applies to NAWIs, whereas this software guide applies to AWIs.**

The specific requirements of this chapter are based on Annex MI-006 and the normative documents mentioned in 10.6.1 as far as they support the interpretation of MID requirements.

10.6.1 Specific regulations, standards and other normative documents

5 categories of automatic weighing instruments (AWIs) are subject to regulations in MID Annex MI-006:

- Automatic catchweighers (R51)
- Automatic gravimetric filling instruments (R61)
- Discontinuous totalisers (R107)
- Continuous totalisers (belt weighers) (R50)
- Automatic rail weighbridges (R106)

The numbers in brackets refer to the respective OIML recommendations that are normative documents in the sense of the MID. In addition, WELMEC has issued the WELMEC Guide 2.6 that supports the testing of automatic catchweighers.

There is one category of AWIs that is not governed by the MID:

- Automatic instruments for weighing road vehicles in motion (R134)

AWIs of all categories may be realised as type P or type U, and all extensions could be relevant for each category.

However, of these 6 categories, only **discontinuous totalisers** and **continuous totalisers** (belt weighers) have been identified as requiring instrument specific software requirements (see 10.6.3). The reason is that the measurement is cumulative over a relatively long period of time and cannot be repeated if a significant fault occurs.

10.6.2 Technical description

10.6.2.1 Hardware Configuration

A discontinuous totaliser is a totalising hopper weigher that determines the mass of a bulk product (eg. grain) by dividing it into discrete loads. The system usually comprises of one or more hoppers supported on load cells, power supply, electronic controls and indicating device.

A continuous totaliser is a belt weigher that measures the mass of a product as the belt passes over a load cell. The system usually comprises of a conveyor belt, rollers, load receptor supported on load cells, power supply, electronic controls and indicating device. There will be a means for adjusting the tension of the belt.

10.6.2.2 Software Configuration

This is specific to each manufacturer but would normally expect to follow the recommendations given in the main body of this guide.

10.6.2.3 Measuring Principle

In the case of a discontinuous totaliser the bulk product is fed into a hopper and weighed. The mass of each discrete load is determined in sequence and summed. Each discrete load is then delivered to bulk.

In the case of a continuous totaliser the mass is continually measured as the product passes over the load receptor. Measurements are made in discrete units of time that depend on the belt speed and the force on the load receptor. There is no deliberate subdivision of the product or interruption of the conveyor belt as with a discontinuous totaliser. The total mass is an integration of the discrete samples. It should be noted that the load receptor could use strain gauge load cells or other technologies such as vibrating wire.

10.6.2.4 Defects

Joints in the belt may generate shock effects, which can lead to erroneous events when zeroing. In the case of discontinuous totalisers, single or all weighing results of discrete loads may get lost before being summed up.

10.6.3 Specific software requirements (Discontinuous and Continuous Totalisers)

MID Annex MI-006, Chapter IV, Section 8, and Chapter V, Section 6 deal with electromagnetic disturbances. There is a need to interpret these requirements for software controlled instruments because the detection of a disturbance (fault) and subsequent recovery are only possible through the co-operation of specific hardware parts and specific software. From the software point of view, it makes no difference what the reason of a disturbance was (electromagnetic, electrical, mechanical etc); the recovery procedures are all the same.

Risk Class B	Risk Class C	Risk Class D
I6-1: Fault Detection *The software shall detect that normal processing is disturbed.*		
Specifying Notes: On detection of a fault: a. The cumulative measurement and other relevant legal data shall be automatically saved to non-volatile storage (see Requirement I6-2). b. The hopper weigher or belt weigher shall be stopped. c. An alarm shall be sounded.		
Required Documentation: A brief description of what is checked, what is required to trigger the fault detection process, what action is taken on the detection of a fault.		
Validation Guidance: *Checks based on documentation:* • Check whether the realisation of fault detection is appropriate. *Functional checks:* • If possible: simulate certain hardware faults and check whether they are detected and reacted upon by the software as described in the documentation.		
Acceptable solution: A hardware watchdog is reset by a cyclically processed microprocessor subroutine in order to inhibit the firing of the watchdog. Before resetting, the subroutine checks the health of the system eg. whether all metrologically relevant subroutines have been processed during the last interval. If any function has not been processed or - in the worst case - the microprocessor hang in an arbitrary endless loop, the reset of the watchdog doesn’t happen and it fires after a certain time span.		

Risk Class B	Risk Class C	Risk Class D

I6-2: Back-up Facilities

There shall be a fall-back facility that allows the back-up of essential data such as. measurement values and the current status of the process in case of a disturbance.

Specifying Notes:

a. The state characteristics and important data shall be stored in a non-volatile storage.

b. Periodic backing up is acceptable if a controlled storage facility is not available due to hardware or functional constraints. However, the storage intervals must be small so that the discrepancy between the current and saved values is small compared to the maximum permissible error.

c. The back-up facilities should normally include appropriate wake-up facilities in order that the weighing system, including its software, does not get into an indefinite state by a disturbance.

Required Documentation:

A brief description of which data is backed up and when this occurs. Calculation of the maximum error that can occur for cumulative values if a cyclical back-up is realised.

Validation Guidance:

Checks based on documentation:

- Check whether all legally relevant data are saved in case of a disturbance.

Functional checks:

- Check by simulating a disturbance whether back-up mechanism works as described in the documentation.

Acceptable solution:

A hardware watchdog fires when it is not cyclically reset. This alarm actuates an interrupt in the microprocessor. The assigned interrupt routine collects measurement values, state values and other relevant data and stores them in a non-volatile storage eg. an EEPROM.

Note: It is assumed that the watchdog interrupt has highest interrupt priority and can dominate any normal processing or any arbitrary endless loop, ie. the program control always jumps to the interrupt routine if the watchdog fires.

10.6.4 Examples of legally relevant functions and data

Table 10.6.1: Examples of legally relevant, device-specific and type-specific functions and data (DF, DD, TF, TD) for AWIs in comparison with those of non-automatic weighing instruments (R76). VV indicates variable values.

Functions/data	**Type**	**OIML Recommendation No**						
		50	**51 (X)**	**51 (Y)**	**61**	**76**	**106**	**107**
Weight calculation	TF, TD	X	X	X	X	X	X	X
Stability analysis	TF, TD		X	X	X	X	X	X
Price calculation	TF, TD			X		X		
Rounding algorithm for price	TF, TD			X		X		
Span (sensitivity)	DD	X	X	X	X	X	X	X
Corrections for non-linearity	DD (TD)	X	X	X	X	X	X	X
Max, Min, e, d	DD (TD)	X	X	X	X	X	X	X
Units of measurement (eg. g, kg)	DD (TD)	X	X	X	X	X	X	X
Weight value as displayed (rounded to multiples of e or d)	VV	X		X		X	X	X
Tare, preset tare	VV		X	X	X	X	X	
Unit price, price to pay	VV			X		X		X
Weight value in internal resolution	VV	X	X	X	X	X	X	X
Status signals (eg. zero indication, stability of equilibrium)	TF	X	X	X	X	X	X	X
Comparison of actual weight vs. preset value	TF		X		X			
Automatic printout release, eg. at interruption of automatic operation	TF	X						X
Warm-up time	TF (TD)	X	X	X	X	X	X	X
Interlock between functions	TF		X	X				
eg. zero setting/tare			X	X	X	X		
automatic/non-automatic operation,							X	
zero-setting/totalizing		X						X
Record of access to dynamic setting	TF (VV)		X	X				
Maximum rate of operation/range of operating speeds (dynamic weighing)	DD (TD)	X	X	X	X		X	X
(Product)-Parameters for dynamic weight calculation	VV		X	X			X	
Preset weight value	VV		X		X			
Width of adjustment range	DD (TD)		X	X				
Criterion for automatic zero-setting (eg. time interval, end of weighing cycle)	DD (TD)		X	X	X		X	X
Minimum discharge, rated minimum fill	DD				X			X
Limiting value of significant fault (if not 1e or 1d)	DD (TD)	X			X			
Limiting value of battery power	DD (TD)	X	X	X	X	X	X	X

The marked functions and parameters are likely to occur on the various types of weighing instruments. If one of them is present, it has then to be treated as "legally relevant". The table is however not meant as an obligatory list indicating that any function or parameter mentioned has to be realised in each instrument.

10.6.5 Other aspects

None

10.6.6 Assignment of risk class

For the present, according to the decision of the responsible WELMEC Working Group (24th WG2 meeting, 22/23 January 2004) risk class "B" shall be generally applied to all categories of AWIs regardless of the type (P or U).

However, as a result of the WG7 questionnaire (2004), the following differentiation with regard to type P and U instruments, and to discontinuous and continuous totalising instruments seems appropriate and should be discussed again in WELMEC WG2:

- risk class B for type P instruments (except totalisers)
- risk class C for totalisers of type P, and for other AWIs of type U
- risk class D for totalisers of type U

10.7 Taximeters

Taximeters are subject to regulations in MID. The specific requirements are in Annex MI-007. Neither these specific requirements nor any normative documents have yet been taken into consideration.

10.7.1 - 10.7.5 will be filled in if considered necessary in the future. In that case the specific issue of reliability of the velocity signal should be taken into account.

10.7.6 Assignment of risk class

For the present, according to the result of the WELMEC WG7 questionnaire (2004) and subject to future decisions of the responsible WELMEC Working Group, the following risk class should be applied if software examinations based on this guide are carried out for (software-controlled) taximeters:

- Risk class C for type P instruments
- Risk class D for type U instruments

10.8 Material Measures

Material measures are subject to regulations in MID. The specific requirements are in Annex MI-008.

Subject to future developments and decisions material measures in the sense of MID Annex MI-008 are not considered ~~as~~to be software-controlled measuring instruments. Thus, for the present, this software guide does not apply to material measures.

10.9 Dimensional Measuring Instruments

Dimensional Measuring Instruments are subject to regulations in MID. The specific requirements are in Annex MI-009. Neither these specific requirements nor any normative documents have yet been taken into consideration.

10.9.1 - 10.9.5 will be filled in if considered necessary in the future.

10.9.6 Assignment of risk class

For the present, according to the result of the WELMEC WG7 questionnaire (2004) and subject to future decisions of the responsible WELMEC Working Group, the following risk class should be applied if software examinations based on this guide are carried out for (software-controlled) dimensional measuring instruments:

- Risk class B for type P instruments
- Risk class C for type U instruments

10.10 Exhaust Gas Analysers

Exhaust Gas Analysers are subject to regulations in MID. The specific requirements are in Annex MI-010. Neither these specific requirements nor any normative documents have yet been taken into consideration.

10.10.1 - 10.10.5 will be filled in if considered necessary in the future.

10.10.6 Assignment of risk class

For the present, according to the result of the WELMEC WG7 questionnaire (2004) and subject to future decisions of the responsible WELMEC Working Group, the following risk class should be applied if software examinations based on this guide are carried out for (software-controlled) exhaust gas analysers:

- Risk class B for type P instruments
- Risk class C for type U instruments

11 Definition of Risk Classes

11.1 General principle

The requirements of this guide are differentiated according to risk classes. Each measuring instrument must be assigned to a risk class because the particular software requirements to be applied are governed by the risk class the instrument belongs to. A risk class is defined by the combination of the appropriate levels required for software protection, software examination and software conformity. Three levels, low, middle and high are introduced for each of these categories.

11.2 Description of levels for protection, examination and conformity

The following definition are used for the corresponding levels.

Software protection levels

Low: No particular protection measures against intentional changes are required.

Middle: The software is protected against intentional changes made by using easily-available and simple common software tools (e.g. text editors).

High: The software is protected against intentional changes made by using sophisticated software tools (debuggers and hard disc editors, software development tools, etc).

Software examination levels

Low: Standard type approval functional testing of the instrument is performed. No extra software testing is required.

Middle: In addition to the low level, the software is examined on the basis of its documentation. The documentation includes the description of the software functions, parameter description, etc. Practical tests of the software-supported functions (spot checks) may be carried out to check the plausibility of documentation and the effectiveness of protection measures.

High: In addition to the middle level, an in-depth test of the software is carried out, usually based on the source code.

Software conformity levels

Low: The functionality of the software implemented for each individual instrument is in conformity with the documentation approved.

Middle: In addition to the conformity level "low", depending on the technical features, parts of the software shall be defined as fixed at type approval, i.e. unalterable without NB approval. The fixed part shall be identical in every individual instrument.

High: The software implemented in the individual instruments is completely identical to the approved one.

11.3 Derivation of risk classes

Out of the 27 theoretically possible level permutations, only 4 or at the utmost 5 are of practical interest (risk classes B, C, D and E, eventually F). They cover all of the instrument classes falling under the regulation of MID. Moreover, they provide a sufficient window of opportunity for the case of changing risk evaluations. The classes are defined in the table below.

Risk Class	Software Protection	Software Examination	Degree of Software Conformity
A	*low*	*low*	*low*
B	*middle*	*middle*	*low*
C	*middle*	*middle*	*middle*
D	*high*	*middle*	*middle*
E	*high*	*high*	*middle*
F	*high*	*high*	*high*

Table: Definition of risk classes

11.4 Interpretation of risk classes

Risk class A: It is the lowest risk class at all. No particular measures are required against intentional changes of software. Examination of software is part of the functional testing of the device. Conformity is required on the level of documentation. It is not expected that any instrument is classified as a risk A instrument. However, by introducing this class, the corresponding possibility is held open.

Risk class B: In comparison to risk class A, the protection of software is required on the middle level. Correspondingly, the examination level is uprated to the middle level. The conformity remains unchanged in comparison to risk class A.

Risk class C: In comparison to risk class B, the conformity level is raised to "middle". This means, parts of the software may declared as fixed at type approval. The rest of the software is required to be conform on the functional level. The levels of protection and examination remain unchanged in comparison to risk class B.

Risk class D: The significant difference in comparison to risk class C is the raising of the protection level to “high”. Since the examination level remains unaffected at “middle”, sufficiently informative documentation must be provided to show that the protection measures taken are appropriate. The conformity level remains unchanged in comparison to risk class C.

Risk class E: In comparison to risk class D, the examination level is upgraded to “high”. The levels of protection and conformity remain unchanged.

Risk class F: The levels with respect to all aspects (protection, examination and conformity) are set to “high”. Like risk class A, it is not expected that any instrument is classified as a risk F instrument. However, by introducing this class, the corresponding possibility is held open.

12 Pattern For Test Report (Including Checklists)

This is a pattern for a test report, which consists of a main part and two annexes. The main part contains general statements on the object under test. The annex 1 consists of two checklists to support composition the appropriate parts of the guide to be applied. The annex 2 consists of the specific checklists for the respective technical parts of the guide. They are used as an aid for manufacturer and examiner to prove that they have considered all applicable requirements.

Test report no XYZ122344

Flow meter Dynaflow model DF101

Validation of Software

(n annexes)

Commission

The Measuring Instruments Directive (MID) gives the essential requirements for certain measuring instruments used in the European Union. The software of the measuring instrument was validated to show conformance with the essential requirements of the MID.

The validation was based on the report WELMEC MID Software Requirements Guide WELMEC Guide 7.x), where the essential requirements are interpreted and explained for software. This report describes the examination of software needed to state conformance with the MID.

Client

Dynaflow
P.O. Box 1120333
100 Reykjavik
Iceland
Reference: Mr Bjarnur Sigfridson

Test Object

The Dynaflow flow meter DF100 is a measuring instrument intended to measure flow in liquids. The intended range is from 1 l/s up to 2000 l/s.

The basic functions of the instrument are:
- measuring of flow in liquids
- indication of measured volume
- interface to transducer

According to the WELMEC Guide 7.x, the flow meter is described as follows:
- a built-for-purpose Measuring instrument (an embedded system)
- long-term storage of legally relevant data

The flow meter DF100 is an independent instrument with a transducer connected. The transducer is fixed to the instrument and cannot be disconnected. The measured volume is indicated on a display. No communication with other devices is possible.

The embedded software of the measuring instrument was developed by Dynaflow,
P.O. Box 1120333, 100 Reykjavik, Iceland.

The version of the software validated is V1.2c. The source code comprises following files:

main.c	12301 byte	23 Nov 2003
int.c	6509 byte	23 Nov 2003
filter.c	10897 byte	20 Oct 2003
input.c	2004 byte	20 Oct 2003
display.c	32000 byte	23 Nov 2003
Ethernet.c	23455 byte	15 June 2002
driver.c	11670 byte	15 June 2002
calculate.c	6788 byte	23 Nov 2003

~~The software is protected against modification by a checksum. (?? Soll das stehen bleiben,. Es wäre ein heraus genommenes Element aus der Lösung)~~

The validation has been supported by following document from the manufacturer:
DF 100 User Manual
DF 100 Maintenance Manual
Software description DF100 (internal design document, dated 22 Nov 2003)
Electronic circuit diagram DF100 (drawing no 222-31, date 15 Oct 2003)

The final version of the test object was delivered to National Testing & Measurement Laboratory on 25 November 2003.

Examination Procedure~~Performance~~ ~~(?? Ist das die richtige Überschrift?)~~

The validation has been performed according to the WELMEC MID Software Requirements Guide (version 1.0, downloaded at www.welmec.org).

The validation was performed between 1 November and 23 December 2003. A design review was held on 3 December by Dr K. Fehler at Dynaflow head office in Reykjavik. Other validation work has been carried out at the National Testing & Measurement Lab by Dr K Fehler and M. S. Problème.

Following requirements have been validated:
- Specific requirements for embedded software for a built-for-purpose measuring instrument (type p)
- Extension I: Long-term storage for legally relevant data

Checklist for the selection of the configuration is found in annex 1 to this report.

Risk class X has been applied to this instrument.

Following validation methods have been applied:
- identification of the software
- completeness of the documentation
- examination of the operating manual
- functional testing
- software design review (??)

- review of software documentation
- data flow analysis (??)
- simulation of input signals

Result

Following requirements of the MID-Software Requirements Guide have been validated without finding faults:
P1 P2 P3, P5, P6, P7
L1, L2, L3, L4, L5, L6, L7
(Requirement P4is considered to be non-applicable.)
Checklists for the P-requirements are found in annex 2.1 of this report.
Checklists for the L-requirements are found in annex 2.2 of this report.

Two commands which were not initially described in the operators manual were found. The two commands have been included in the operators manual dated 10 December 2003.

A software fault which limited the month of February to 28 days also in leap year was found in software package V1.2b. This has been corrected in V1.2c.

The software of the Dynaflow DF100 V1.2c fulfils the essential requirements of the Measuring Instruments Directive.

The result applies to the tested item only.

National Testing & Measurement Lab
Software Department

Dr. K.E.I.N. Fehler	M. S.A.N.S Problème
Technical manager	Technical Officer

Date: 23 December 2003

13 Annex 1: Checklists to support the selection of the appropriate requirement Sets

The first checklist supports the user to decide which of basic configuration P or U applies. For the instrument under test.

Decision on Instrument Type				
Reference		P	U	Remarks
?	Is the entire application software constructed for the measuring purpose?	Y	N	
?	If there is general-purpose software, is it accessible by or visible to the user?	N	Y	
?	Is the user prevented from accessing the operating system if it is possible to switch to an operating mode not subject to legal control?	Y	N	
?	Are the implemented programs and the software environment invariable (apart from updates)?	Y	N	
?	Are there any means for programming?	N	Y	
Tick the empty boxes, as appropriate				

~~Decision on Instrument Type~~					
~~Reference~~		~~Yes~~	~~No~~	~~Not Applicable~~	~~Remarks~~
	~~Is the entire software constructed for the measuring purpose and if there is general purpose software, is it not accessible or visible for the user?~~				
	~~If switching to an operating mode not subject to legal control is possible: Is the user prevented to access the operating system?~~				
	~~Are the implemented programs and the software environment invariable and are there no means for programming and if loading of legally relevant software is intended: Is the related requirement set D considered?~~				
~~The instrument is of Type U, if at least one question above was answered with NO, other wise of type P!~~					

~~The above table is difficult to understand because it contains too many double-negative questions in order to force a *yes or no* answer. I think that the third row applies to both P and U because software can be downloaded to both.~~

~~I think that the main difference between P and U is that U can contain non-legally relevant applications that the user can easily switch to – see the amended table below.~~

~~Decision on Instrument Type~~				
~~Reference~~		~~P~~	~~U~~	~~Remarks~~
	~~Can the instrument contain non-legally relevant application software that the user may switch to?~~	~~N~~	~~Y~~	

	~~Can the user switch to an operating mode not subject to legal control?~~	~~N~~	~~Y~~	
	~~Tick the empty boxes, as appropriate~~			

The second checklist supports to decide which of the IT configuration applies for the instrument under test.

Decision on Required Extensions					
Req. Extension		**Yes**	**No**	**Not Applicable**	**Remarks**
L	Does the device have the ability to store the measurement data either on an integrated storage or on a remote or removable storage?				
T	Does the device have interfaces for transmission of data to devices subject to legal control OR is the device receiving data from another device subject to legal control?				
S	Are there software parts with functions not subject to legal control AND are these software parts desired to be changed after type approval?				
D	Is loading of software possible or desired?				
Consider the required extension for each question answered with YES!					

14 Annex 2: Specific checklists for the respective technical parts

1) Checklist, basic requirements for type P instrument

Checklist for Type P Requirements						
Requirement	Testing procedures ??		Passed	Failed	Not Applicable	Remarks*
P1		Does the required manufacturer documentation fulfil the requirement P1(a-g)?				
P2		Is a software identification realised as required in P2?				
P3		Are commands entered via the user interface prevented from inadmissibly influencing the legally relevant software and measurement data?				
P4		Are commands input via non-sealed communication interfaces of the instrument prevented from inadmissibly influencing the legally relevant software and measurement data?				
P5		Are legally relevant software and measurement data protected against accidental or unintentional changes?				
P6		Are legally relevant software secured against the inadmissible modification, loading or swapping of hardware memory?				
P7		Are parameters that fix legally relevant characteristics of the measuring instrument secured against unauthorised modification?				
** Explanations are needed if there are deviations from software requirements.*						

2) Checklist, basic requirements for type U instrument

Checklist for Type U Requirements						
Requirement	Testing procedures ??		Passed	Failed	Not Applicable	Remarks*
U1		Does the required manufacturer's documentation fulfil the requirement U1(a-h)?				
U2		Is a software identification realised as required in U2?				
U3		Are commands entered via the user interface prevented from inadmissibly influencing the legally relevant software and measurement data?				
U4		Is it prevented that commands inputted via non-sealed communication interfaces of the instrument inadmissibly influence the legally relevant software and measurement data?				
U5		Are legally relevant software and measurement data protected against accidental or unintentional changes?				
U6		Are legally relevant software secured against inadmissible modification?				
U7		Are legally relevant parameters secured against unauthorised modification?				
U8		Are means employed to ensure the authenticity of the legally relevant software and are the authenticity of the results that are presented guaranteed?				

U9		Is the legally relevant software designed in such a way that other software does not inadmissibly influence it?				
** Explanations are needed if there are deviations from software requirements.*						

3) Checklist, specific requirements extension L

Checklist for Requirements of Extension L						
Requirement	**Testing procedures ??**		**Passed**	**Failed**	**Not Applicable**	**Remarks***
L1		Do the stored measurement data contain all relevant information necessary to reconstruct an earlier measurement?				
L2		Are stored data protected against accidental and unintentional changes?				
L3		Are the stored measurement data protected against intentional changes carried out by *simple common software tools* (for risk classes B&C) or by *special sophisticated software tools* (for risk classes D&E)?				
L4		Are the stored measurement data capable of being authentically traced back to the measurement that generated them?				
L5		B&C) Are keys treated as legally relevant data and kept secret and protected against compromise by *simple software tools*?				
		D&E) Are keys and accompanying data treated as legally relevant data and kept secret and protected against compromise by sophisticated software tools? Are Appropriate methods equivalent to electronic payment used? Is user able to verify the authenticity of the public key?				
L6		Does the software used for verifying stored measurement data sets display or print the data, check the data for changes, and warn if a change has occurred? Are there means to prevent data detected as having been corrupted to be used?				
L7		Are the measurement data stored automatically when the measurement is concluded?				
L8		Does the long-term storage have a capacity which is sufficient for the intended purpose?				
** Explanations are needed if there are deviations from software requirements.*						

4) Checklist, specific requirements extension T

Checklist for Requirements of Extension T						
Requirement	**Testing procedures ??**		**Passed**	**Failed**	**Not Applicable**	**Remarks***
T1		Do transmitted data contain all relevant information necessary to present or further process the measurement result in the receiving module?				
T2		Are transmitted data protected against accidental and unintentional changes?				
T3		Are legally relevant transmitted data protected against intentional changes carried out by *simple common software tools* (for risk classes B&C) or by *special sophisticated software tools* (for risk classes D&E)?				
T4		Is it possible for the program that receives transmitted relevant data to verify their authenticity and to assign the measurement values to a particular measurement?				
T5		B&C) Are keys treated as legally relevant data and kept secret and protected against compromise by *simple software tools*?				
		D&E) Are keys and accompanying data treated as legally relevant data and kept secret and protected against compromise by sophisticated software tools? Are Appropriate methods equivalent to electronic payment used? Is user able to verify the authenticity of the public key?				
T6		Are data that have been detected as having been corrupted, prevented from being used?				
T7		Is it ensured that the measurement is not inadmissibly influenced by a transmission delay?				
T8		Is it ensured that no measurement data get lost if network services become unavailable,?				
** Explanations are needed if there are deviations from software requirements.*						

5) Checklist, specific requirements extension S

Checklist for Requirements of Extension T						
Requirement	**Testing procedures ??**		**Passed**	**Failed**	**Not Applicable**	**Remarks***
S1		Does the software that is subject to legal control contain all legally relevant software and parameters?				
S2		Is it ensured that additional information generated by the legally non relevant software part, shown on a display or printout, cannot be confused with the information that originates from the legally relevant part?				
S3		Is the data exchange between the legally relevant and non-legally relevant software performed via a protective software interface that comprises controls the interactions and data flow?				
** Explanations are needed if there are deviations from software requirements.*						

6) Checklist, specific requirements extension D

Checklist for Requirements of Extension T						
Requirement	**Testing procedures ??**		**Passed**	**Failed**	**Not Applicable**	**Remarks**[*]
D1		Is downloading and the subsequent installation of software automatic? Is it ensured that the software protection environment is at the approved level on completion?				
D2		Are means employed to guarantee that the downloaded software is authentic, and to indicate that the downloaded software has been approved by an NB?				
D3		Are means employed to guarantee that the downloaded software has not been inadmissibly changed during download?				
D4		Is it guaranteed by appropriate technical means that downloads of legally relevant software are adequately traceable within the instrument for subsequent controls?				
D5		Is it guaranteed by technical means that software may only be loaded with the explicit consent of the user or owner of the measuring instrument, as appropriate?				
* *Explanations are needed if there are deviations from software requirements.*						

15 Cross Reference for MID-Software Requirements to MID Articles and Annexes

(Related MID Version: Directive 2004/22/EC, 31 March 2004 (final), related MID-Software Guide Version: 0.055, 31 August 2004)

15.1 Given software requirement, reference to MID

Requirement		MID	
No	Denotation	Article / Annex No (AI = Annex I)	Denotation
	Basic Guide P		
P1	Manufacturer's Documentation	AI-9.3 AI-12 Article 10	Information to be borne by and to accompany the instrument Conformity Evaluation Technical Documentation
P2	Software Identification	AI-7.6 AI-8.3	Suitability Protection against corruption
P3	Influence via User Interface	AI-7.1	Suitability
P4	Influence via communication Interface	AI-7.1 AI-8.1	Suitability Protection against corruption
P5	Protection Against Accidental or Unintentional Changes	AI-7.1, AI-7.2 AI-8.4	Suitability Protection against corruption
P6	Protection Against Intentional Changes	AI-7.1 AI-8.2, AI-8.3, AI-8.4	Suitability[1] Protection against corruption
P7	Parameter Protection	AI-7.1 AI-8.2, AI-8.3, AI-8.4	Suitability Protection against corruption
	Basic Guide U		
U1	Manufacturer's Documentation	AI-9.3 AI-12 Article 10	Information to be borne by and to accompany the instrument Conformity Evaluation Technical Documentation
U2	Software Identification	AI-7.6 AI-8.3	Suitability Protection against corruption
U3	Influence via user interfaces	AI-7.1	Suitability
U4	Influence via Communication Interface	AI-7.1 AI-8.1	Suitability Protection against corruption
U5	Protection against accidental or unintentional changes	AI-7.1, AI-7.2 AI-8.4	Suitability Protection against corruption
U6	Protection against Intentional Changes	AI-7.1 AI-8.2, AI-8.3, AI-8.4	Suitability Protection against corruption
U7	Parameter Protection	AI-7.1 AI-8.2, AI-8.3, AI-8.4	Suitability Protection against corruption
U8	Software authenticity and Presentation of Results	AI-7.1, AI-7.2, AI-7.6 AI-8.3 AI-10.2, AI-10.3, AI-10.4	Suitability Protection against corruption Indication of result
U9	Influence of other software	AI-7.6	Suitability
	Extension L		
L1	Completeness of stored data	AI-7.1 AI-8.4 AI-10.2	Suitability Protection against corruption Indication of result
L2	Protection against accidental or unintentional changes	AI-7.1, AI-7.2 AI-8.4	Suitability Protection against corruption

[1] Note: As regards contents, paragraph 7.1 of MID-Annex I is not an issue of "Suitability" but of "Protection against corruption" (Paragraph 8)

Requirement		MID	
No	**Denotation**	**Article / Annex No** (AI = Annex I)	**Denotation**
L3	Integrity of data	AI-7.1 AI-8.4	Suitability Protection against corruption
L4	Authenticity of stored data	AI-7.1 AI-8.4 AI-10.2	Suitability Protection against corruption Indication of result
L5	Confidentiality of keys	AI-7.1 AI-8.4	Suitability Protection against corruption
L6	Retrieval of stored data	AI-7.2 AI-10.1, AI-10.2, AI-10.3, AI-10.4	Suitability Indication of result
L7	Automatic storing	AI-7.1 AI-8.4	Suitability Protection against corruption
L8	Storage capacity and continuity	AI-7.1	Suitability
Lx	All of Extension L	AI-11.1	Further processing of data to conclude the trading transaction
	Extension T		
T1	Completeness of transmitted data	AI-7.1 AI-8.4	Suitability Protection against corruption
T2	Protection against accidental changes	AI-7.1, AI-7.2 AI-8.4	Suitability Protection against corruption
T3	Integrity of data	AI-7.1 AI-8.4	Suitability Protection against corruption
T4	Authenticity of transmitted data	AI-7.1 AI-8.4	Suitability Protection against corruption
T5	Confidentiality of keys	AI-7.1 AI-8.4	Suitability Protection against corruption
T6	Handling of corrupted data	AI-7.1 AI-8.4	Suitability Protection against corruption
T7	Transmission delay	AI-7.1 AI-8.4	Suitability Protection against corruption
T8	Availability of transmission services	AI-7.1 AI-8.4	Suitability Protection against corruption
	Extension S		
S1	Realisation of software separation	AI-7.6, AI-10.1	Suitability Indication of result
S2	Mixed indication	AI-7.1, AI-7.2, AI-7.6 AI-10.2	Suitability Indication of result
S3	Protective software interface	AI-7.6	Suitability
	Extension D		
D1	Download mechanism	AI-8.2, AI-8.4	Protection against corruption
D2	Authentication of downloaded software	AI-7.6 AI-8.3, AI-8.4 AI-12	Suitability Protection against corruption Conformity evaluation
D3	Integrity of downloaded software	AI-7.1, AI-8.4	Suitability Protection against corruption
D4	Traceability of legally relevant Software Download	AI-7.1, AI-7.6 AI-8.2, AI-8.3 AI-12	Suitability Protection against corruption Conformity evaluation
D5	Download Consent	AI-7.1, AI-7.6	Suitability
	Extension I (Instrument specific Software Requirements)		
I1-1, I2-1, I3-1, I4-1	Fault Detection	AI-6 MI-001-7.1, MI-002-3.1, MI-003-4.3.1, MI-004-4	Reliability Specific Requirements for Utility Meters

Requirement		MID	
No	**Denotation**	**Article / Annex No** (AI = Annex I)	**Denotation**
I1-2, I2-2, I3-2, I4-2	Back-up Facilities	AI-6 MI-001-7.1, MI-002-3.1, MI-003-4.3.1, MI-004-4	Reliability Specific Requirements for Utility Meters
I1-3, I2-3, I3-3, I4-3	Wake-up Facilities and Restoring	AI-6 MI-001-7.1, MI-002-3.1, MI-003-4.3.1, MI-004-4	Reliability Specific Requirements for Utility Meters
I1-4, I2-4, I3-4, I4-4	Internal Resolution	MI-002-5.3, MI-003-5.2	Specific Requirements for Utility Meters
I1-5, I2-5, I3-5, I4-5	Inhibit resetting of cumulative measurement values	AI-8.5	Protection against corruption
I1-6, I2-6, I3-6, I4-6	Indication for the customer	AI-7.2 AI-10.5	Suitability Indication of result
I2-7	Acc. Sol. for monitoring of battery lifetime	MI-002-5.2	Specific Requirements for Gas Meters
I2-8	Acc. Sol. for monitoring of gas volume converters	MI-002-9.1	Specific Requirements for Gas Meters
I2-9	Test element	MI-002-5.5	Specific Requirements for Gas Meters
I6-1	Fault detection	MI-006-IV, MI-006-V	Discontinuous and continuous Totalisers
I6-2	Back-up facilities	MI-006-IV, MI-006-V	Discontinuous and continuous Totalisers

15.2 Interpretation of MID Articles and Annexes by MID-Software Requirements

MID			Software Guide
Article / Annex No (AI = Annex I)	**Denotation**	**Comment**	**Requirement No**
	Article Part		
1, 2, 3		No specific software relevance	
4(b)	Definitions, Arrangement of sub-assemblies	Transmission of legally relevant data ... Basic Guides applicable to sub-assemblies	T.x P, U
5 to 9		No specific software relevance	
10	Technical documentation	Documentation of design, manufacture and operation. Enable assessment of conformity. General description of the instrument. Description of electronic devices with drawings, flow diagrams of the logic, general software information. Location of seals and markings. Conditions for compatibility with interfaces and sub-assemblies.	P1, U1
11 to 27		No specific software relevance	

MID			Software Guide
Article / Annex No (AI = Annex I)	Denotation	Comment	Requirement No
	Annex I		
AI-1 to AI-5		No specific software relevance	
AI-6	Reliability	Fault detection, back-up, restoring, restart	I1-1 to I1-3, I2-1 to I2-3, I3-1 to I3-3, I4-1 to I4-3, I6-1 to I6-2
AI-7	Suitability	No features to facilitate fraudulent use; minimal possibilities for unintentional misuse.	P3 - P7, U3 - U8, L1 – L5, L7, L8 T1 – T8, S2, D3, D4
AI-8	Protection against corruption		
AI-8.1		No influences by the connection of other devices.	P4, U4
AI-8.2		Securing; evidence of intervention	P6, P7, U6, U7, D1, D4
AI-8.3		Identification of software; evidence of intervention	P2, P6, P7, U2, U6, U7, U8, D2, D4
AI-8.4		Protection of stored or transmitted data	P5 - P7, U5 - U7, L1 - L5, T1 - T8 D1 - D3
AI-8.5		No reset of cumulative registers	I1-5, I2-5, I3-5, I4-5
AI-9	Information to be borne by and to accompany the instrument		
AI-9.1		Measuring capacity (rest of items not relevant for software)	L8
AI-9.2		No specific software relevance	
AI-9.3		Instructions for installation, ..., conditions for compatibility with interface, sub-assemblies or measuring instruments.	P1, U1
AI-9.4 to AI-9.8		No specific software relevance	
AI-10	Indication of result		
AI-10.1		Indication by means of a display or hard copy.	U8, L6, S2
AI-10.2		Significance of result, no confusion with additional indications.	U8, L1, L4, L6, S2
AI-10.3		Print or record easily legible and non-erasable.	U8, L6, S2
AI-10.4		For direct sales: presentation of the result to both parties.	U8, S2
AI-10.5		For utility meters: display for the customer.	I1-6, I2-6, I3-6, I4-6
AI-11	Further processing of data to conclude the trading transaction		
AI-11.1		Record of measurement results by a durable means.	L1 - L8
AI-11.2		Durable proof of the measurement result and information to identify a transaction.	L1, L6

MID			Software Guide
Article / Annex No (AI = Annex I)	Denotation	Comment	Requirement No
AI-12	Conformity evaluation	Ready evaluation of the conformity with the requirements of the Directive.	P1, P2, U1, U2, D2, D4
	Annexes A1 to H1		
A1 to H1		No requirements to features of instruments	
	Annex MI-001		
MI-001-1 to MI-001-6		No specific software relevance	
MI-001-7.1.1, MI-001-7.1.2	Electromagnetic immunity	Fault detection Back-up facilities Wake-up facilities and restoring	I1-1 to I1-3
MI-001-7.1.3 to MI-001-9		No specific software relevance	
	Annex MI-002		
MI-002-1 to MI-002-2		No specific software relevance	
MI-002-3.1	Electromagnetic immunity	Fault detection Back-up facilities Wake-up facilities and restoring	I2-1 to I2-3
MI-002-3.1.3 to MI-002-5.1		No specific software relevance	
MI-002-5.2	Suitability	Acceptable solution for monitoring battery lifetime	I2-7
MI-002-5.3	Suitability	Internal resolution	I2-4
MI-002-5.4 to MI-002-8		No specific software relevance	
MI-002-5.5	Suitability	Test element	I2-9
MI-002-5.6 to MI-002-8		No specific software relevance	
MI-002-9.1	Volume conversion devices Suitability	Acceptable solution for monitoring the gas volume converter	I2-8
MI-002-9.2 to MI-002-10		No specific software relevance	
	Annex MI-003		
MI-003-1 to MI-003-4.2		No specific software relevance	
MI-003-4.3	Permissible effect of transient electromagnetic phenomena	Fault detection Back-up facilities Wake-up facilities and restoring	I3-1 to I3-3
MI-003-5.1		No specific software relevance	
MI-003-5.2	Suitability	Internal resolution	I3-4
MI-003-5.3 to MI-003-7		No specific software relevance	
	Annex MI-004		
MI-004-1 to MI-004-4.1		No specific software relevance	
MI-004-4.2	Permissible influences of electromagnetic disturbances	Fault detection Back-up facilities Wake-up facilities and restoring	I4-1 to I4-3
MI-004-4.3 to MI-004-7		No specific software relevance	

MID			Software Guide
Article / Annex No (AI = Annex I)	**Denotation**	**Comment**	**Requirement No**
	Annex MI-005		
	Annex MI-006		
MI-006-IV, MI-006-V	Discontinuous and continuous Totalisers	Fault detection Back-up facilities	I6-1 to I6-2
	Annex MI-007		
	Annex MI-008		
	Annex MI-009		
	Annex MI-010		

16 Index