

## **D31 - 1CD**

Title: General requirements for software-controlled measuring instruments

Notes for 1CD:

- This document implements results from SGs 1, 2 and 3.
- The document also includes all accepted changes proposed in response to 1WD.
- Additionally, this document reflects discussion results from the first PG meeting, especially with respect to the use of components.



## Contents

<b>Foreword.....</b>	<b>4</b>
<b>1 Introduction.....</b>	<b>5</b>
<b>2 Scope and field of application.....</b>	<b>5</b>
<b>3 Terms and definitions .....</b>	<b>6</b>
3.1 General.....	6
3.2 General terminology .....	6
3.3 Abbreviations .....	17
<b>4 Instructions for use of this Document in drafting OIML Recommendations .....</b>	<b>17</b>
<b>5 Risk assessment .....</b>	<b>18</b>
<b>6 Requirements for measuring instruments with respect to the software .....</b>	<b>19</b>
6.1 General.....	19
6.2 General requirements .....	19
6.3 Requirements for specific configurations .....	27
<b>7 Type evaluation .....</b>	<b>49</b>
7.1 Software documentation to be supplied for type evaluation.....	49
7.2 Requirements for the evaluation procedure.....	50
7.3 Verification and evaluation methods.....	52
7.4 Software evaluation procedure.....	60
7.5 Equipment Under Test (EUT).....	60
<b>8 Verification of a measuring instrument.....</b>	<b>61</b>
8.1 General.....	61
8.2 Verification methods, test items.....	61
8.3 Remote Verification.....	62
<b>Annex A Bibliography (Informative) .....</b>	<b>66</b>
<b>Annex B Example of a software test report (Informative).....</b>	<b>69</b>
<b>Annex C Remarks on measurement terminology (Informative) .....</b>	<b>80</b>
<b>Annex D Index.....</b>	<b>83</b>

## Foreword

The International Organisation of Legal Metrology (OIML) is a worldwide, intergovernmental organisation whose primary aim is to harmonise the regulations and metrological controls applied by the national metrological services, or related organisations, of its Member States. The main categories of OIML publications are:

- **International Recommendations (OIML R)**, which are model regulations that establish the metrological characteristics required of certain measuring instruments and which specify methods and equipment for checking their conformity. OIML Member States shall implement these Recommendations to the greatest possible extent;
- **International Documents (OIML D)**, which are informative in nature and which are intended to harmonise and improve work in the field of legal metrology;
- **International Guides (OIML G)**, which are also informative in nature and which are intended to give guidelines for the application of certain requirements to legal metrology; and
- **International Basic Publications (OIML B)**, which define the operating rules of the various OIML structures and systems.

OIML Draft Recommendations, Documents and Guides are developed by Project Groups linked to Technical Committees or Subcommittees which comprise representatives from the Member States. Certain international and regional institutions also participate on a consultation basis. Cooperative agreements have been established between the OIML and certain institutions, such as ISO and the IEC, with the objective of avoiding contradictory requirements. Consequently, manufacturers and users of measuring instruments, test laboratories, etc. may simultaneously apply OIML publications and those of other institutions.

International Recommendations, Documents, Guides and Basic Publications are published in English (E) and translated into French (F) and are subject to periodic revision.

Additionally, the OIML publishes or participates in the publication of **Vocabularies (OIML V)** and periodically commissions legal metrology experts to write **Expert Reports (OIML E)**. Expert Reports are intended to provide information and advice, and are written solely from the viewpoint of their author, without the involvement of a Technical Committee or Subcommittee, nor that of the CIML. Thus, they do not necessarily represent the views of the OIML.

This publication – reference OIML D 31, edition 20<TODO> (E) – was developed by Project Group 4 in the OIML Technical Subcommittee TC 5/SC 2 *Software*. It was approved for final publication by the International Committee of Legal Metrology at its <TODO> meeting in 20<TODO> and will be submitted to the International Conference on Legal Metrology in 20<TODO> for formal sanction.

OIML Publications may be downloaded from the OIML web site in the form of PDF files. Additional information on OIML Publications may be obtained from the Organisation's headquarters:

Bureau International de Métrologie Légale  
11, rue Turgot - 75009 Paris - France  
Telephone: 33 (0)1 48 78 12 82  
Fax: 33 (0)1 42 82 17 27  
E-mail: [biml@oiml.org](mailto:biml@oiml.org)  
Internet: [www.oiml.org](http://www.oiml.org)

# **General requirements for software-controlled measuring instruments**

## **1 Introduction**

The primary aim of this International Document is to provide OIML Technical Committees and Subcommittees with guidance for establishing appropriate requirements for software-related functionalities in measuring instruments covered by OIML Recommendations.

Furthermore, this International Document can provide guidance to OIML Member States in the implementation of OIML Recommendations in their national laws.

## **2 Scope and field of application**

**2.1** This International Document specifies the general requirements applicable to legally relevant software-related functionality and security in measuring instruments and gives guidance for verifying the compliance of an instrument with these requirements.

**2.2** This Document shall be taken into consideration by the OIML Technical Committees and Subcommittees as a basis for establishing specific software requirements and procedures in OIML Recommendations applicable to particular categories of measuring instruments (hereafter termed “relevant Recommendations”).

**2.3** The instructions given in this Document apply only to software-controlled measuring instruments or their components.

*Note 1:* This Document does not cover all the technical requirements specific to software-controlled measuring instruments; these requirements are to be given in the relevant Recommendation, e.g. for weighing instruments, water meters, etc.

*Note 2:* This Document addresses some aspects concerning data, parameter and software security. In addition, national regulations for this area need to be considered.

### 3 Terms and definitions

#### 3.1 General

Some of the definitions used in this Document are in conformity with the International Vocabulary of Metrology - Basic and General Concepts and Associated Terms 3rd Edition (OIML V 2-200:2012 [1]), with the International Vocabulary of Terms in Legal Metrology (OIML V 1:2013 [6]), with the OIML International Document *General requirements for measuring instruments – Environmental conditions* (OIML D 11:2013 [2]) and several ISO/IEC International Standards. For the purpose of this Document, the following definitions and abbreviations apply.

*Note:* Unless stated otherwise, the term certificate refers to the OIML type examination certificate.

#### 3.2 General terminology

##### 3.2.1 audit trail

continuous data containing a time stamped information record of events, e.g. changes in the values of the parameters of a measuring instrument or software updates, or other activities that are legally relevant and which are critical for the metrological characteristics

*Note:* Regarding examples for events logged in an audit trail, see 3.2.19.

adapted from [OIML V 1:2013, 6.05]

##### 3.2.2 authentication

checking of the declared or alleged identity of a user, process, or measuring instrument

*Note:* This may be necessary when checking that downloaded software originates from the owner of the certificate.

##### 3.2.3 authenticity

result of the process of authentication (passed or failed)

##### 3.2.4 built-for-purpose device

device constructed for the specific purpose of a metrological task

*Note 1:* Built-for-purpose devices include devices that may not incorporate an operating system.

*Note 2:* If an operating system is present, it is not directly accessible.

##### 3.2.5 checking facility

facility that is incorporated in a measuring instrument and which enables significant defect to be detected and acted upon

*Note:* “Acted upon” refers to any adequate response by the measuring instrument (luminous signal, acoustic signal, prevention of the measurement process, etc.).

adapted from [OIML V 1:2013, 5.07]

**3.2.6 cloud**

servers that are accessed over the Internet, and the software and databases that run on those servers

**3.2.7 communication interface**

part of an instrument that enables information to be passed between measuring instruments, components of measuring instruments or other external systems

*Note 1:* Communication interfaces can be wired, optical, radio, etc. and they are usually designed to use a specific protocol.

*Note 2:* This definition does not include communication between software modules.

**3.2.8 component**

identifiable part of an instrument that performs a specific function or functions, and that can be separately evaluated according to specific metrological and technical performance requirements as specified in the relevant Recommendation

**3.2.9 cryptographic certificate**

dataset containing the public key belonging to a measuring instrument or a person plus a unique identification of the subject, e.g. serial number of the measuring instrument or name or Personal Identification Number (PIN) of the person, plus a date of expiry

**3.2.10 cryptographic means**

means such as encryption and decryption with the purpose of hiding information from unauthorised persons, or hashes and signatures (see 3.2.16) to ensure integrity and authenticity

**3.2.11 data domain**

location in memory that each program needs for processing data

*Note:* Data domains may belong to one *software module* only, or to several.

**3.2.12 device-specific parameter**

legally relevant parameter with a value that depends on the individual instrument, component and/or module(s) subject to legal control

*Note 1:* Device-specific parameters comprise adjustment parameters (e.g. span adjustment or other adjustments or corrections) and configuration parameters (e.g. maximum value, minimum value, units of measurement, etc.).

*Note 2:* See also 6.2.3.4.

adapted from [OIML V 1:2013, 4.12]

**3.2.13 durability**

ability of the measuring instrument to maintain its performance characteristics over a period of use

[OIML V 1:2013, 5.15]

**3.2.14 dynamic module of legally relevant software**

software module whose functional behaviour depends on predefined device-specific parameters that may change over time during use

**3.2.15 electronic measuring instrument**

measuring instrument intended to measure an electrical or non-electrical quantity using electronic means and/or equipped with electronic parts

*Note:* For the purpose of this Document, auxiliary equipment, provided that it is subject to metrological control, is considered to be part of the measuring instrument.

[OIML D 11:2013, 3.1]

**3.2.16 electronic signature**

software means which is added to software or data with the purpose to verify the origin of software or data, i.e. to prove their authenticity, or to check that the software or data are unchanged, i.e. to prove their integrity

*Note 1:* For electronic signing, a public key system is used in general, i.e. a pair of keys where only one needs to be kept private/secret; the other may be public.

*Note 2:* The private key is used when software or data are secured. The public key is used when software or data are verified before use.

*Note 3:* The verifying instance may require a cryptographic certificate of the securing instance (see 3.2.9) to be sure of the authenticity of the public key.

**3.2.17 error of indication**

indication minus a reference quantity value

*Note:* This reference value is sometimes referred to as a (conventional) true quantity value. See, however, also OIML V 2-200:2012, 2.12, Note 1).

[OIML V 1:2013, 0.04]

**3.2.18 error log**

continuous data file containing an information record of failures or significant defects that have an influence on the legally relevant characteristics of the measuring instrument

**3.2.19 event**

action in which a modification of a measuring instrument parameter, adjustment factor or update of software module is made

*Note:* For the purpose of this Document, events are considered changes in the value of the legally relevant parameters, or a modification or update of the legally relevant software, or other activities that are legally relevant and which may influence the metrological data and/or characteristics.

adapted from [OIML V 1:2013, 6.06]



- 3.2.20 event counter**  
non-resettable counter that increments each time an event occurs
- 3.2.21 executable code**  
digital information installed in the measuring instrument or component (EPROM, hard disk, etc.)  
*Note:* This code is interpreted by the central processing unit (CPU) of the measuring instrument and converted into certain logical, arithmetical, decoding or data transporting operations.
- 3.2.22 fault**  
difference between the error of indication and the intrinsic error of a measuring instrument  
*Note 1:* Principally, a fault is the result of an undesired change of data contained in or flowing through an electronic measuring instrument.  
*Note 2:* From the definition it follows that a “fault” is a numerical value which is expressed either in a unit of measurement or as a relative value, for instance as a percentage.  
[OIML V 1:2013, 5.12]
- 3.2.23 hash function**  
(mathematical) function which maps values from a large (possibly very large) domain into a smaller range  
*Note:* A “good” hash function is such that the results of applying the function to a (large) set of values in the domain will be evenly distributed (and apparently at random) over the range.  
[ISO/IEC 9594-8:2014] [3]
- 3.2.24 integrity (of software, measurement data or parameters)**  
assurance that the software, measurement data or parameters have not been subjected to any unintentional, accidental or inadmissible changes while in use, transfer, storage, repair or maintenance
- 3.2.25 interface**  
shared boundary between two functional units, defined by various characteristics pertaining to the functions, physical interconnections, signal exchanges, and other characteristics of the units, as appropriate  
[ISO 2382-9:1995] [4]
- 3.2.26 interruptible cumulative measurement**  
process of cumulative measurement of the quantity value of a measurand that can be easily and rapidly stopped during normal operation  
*Note 1:* Examples include: a) discontinuous totalising automatic weighing instrument, b) fuel dispenser.  
*Note 2:* See also non-interruptible cumulative measurement (3.2.46).

**3.2.27 legally relevant**

subject to legal control

*Note 1:* If a measuring instrument is under legal control, then the measurement data, software and parameters that are critical for the metrological characteristics, including the metrological functions, securing and protection features, and/or for the completion of the transaction, are also under legal control.

*Note 2:* The relevant Recommendations defines the legally relevant metrological data, characteristics and functions, while giving requirements related to these metrological data, characteristics, functions, securing, protection and the necessary information for the completion of the transaction.

**3.2.28 legally relevant parameter**

parameter of a measuring instrument, component and/or module(s) subject to legal control

*Note:* The following types of legally relevant parameters can be distinguished: type-specific parameters and device-specific parameters.

**3.2.29 legally relevant software**

all software modules of a measuring instrument or component that are subject to legal control

**3.2.30 maximum permissible error (of a measuring instrument)**

extreme value of a measurement error, with respect to a known reference quantity value, permitted by specifications or regulations for a given measurement, measuring instrument, or measuring system

adapted from [OIML V 1:2013, 0.05]

**3.2.31 measuring instrument**

device used for making measurements, alone or in conjunction with one or more supplementary devices

adapted from [OIML V 1:2013, 0.10]

**3.2.32 measurement**

process of experimentally obtaining one or more quantity values that can reasonably be attributed to a quantity

*Note 1:* Measurement does not apply to nominal properties.

*Note 2:* Measurement implies comparison of quantities or counting of entities.

*Note 3:* Measurement presupposes a description of the quantity commensurate with the intended use of a measurement result, a measurement procedure, and a calibrated measuring system operating according to the specified measurement procedure, including the measurement conditions.

*Note 4:* Annex C illustrates the terms and definitions related to the measurement process and their usage in this OIML Document.

adapted from [OIML V 2-200:2012, 2.1]

**3.2.33 measurement data**

data used during the measurement process

*Note:* Measurement data includes the measured quantity value, measurement result relevant data and measurement process data, see Annex C.

**3.2.34 measurement error**

measured quantity value minus a reference quantity value

*Note 1:* The concept of ‘measurement error’ can be used both

- a) when there is a single reference quantity value to refer to, which occurs if a calibration is made by means of a measurement standard with a measured quantity value having a negligible measurement uncertainty or if a conventional quantity value is given, in which case the measurement error is known, and
- b) if a measurand is supposed to be represented by a unique true quantity value or a set of true quantity values of negligible range, in which case the measurement error is not known.

*Note 2:* Measurement implies comparison of quantities or counting of entities.

[OIML V 2-200:2012, 2.16]

**3.2.35 measurement metadata**

metadata related to the measurement process

*Note:* Measurement metadata includes the measured quantity value metadata, measurement result relevant metadata and measurement process metadata.

**3.2.36 measurement process data**

data used during the measurement process to construct the measurement result

*Note:* Examples of measurement process data include values of measurement parameters, values of connection settings or values of session parameters.

**3.2.37 measurement process information**

set of values of qualitative or quantitative variables representing the measurement process

*Note:* Measurement process information includes measurement process data and measurement process metadata.

**3.2.38 measurement process metadata**

metadata related to the measurement process

*Note:* Examples of measurement process metadata include format of the measurement parameters, format of the connection settings or format of the session parameters.

**3.2.39 measurement result**

set of quantity values being attributed to a measurand together with any other available relevant data

*Note 1:* The measurement result relevant data may consist of e.g. measurement uncertainty, date and time of measurement, number of measurement, identification of sensor and in the case where price calculation is part of the legally relevant software, unit price and price to pay.

*Note 2:* The measurement result (including the measured quantity value according to V 2:200:2012) is used for the legally relevant purpose, e.g. conclusion of a transaction.

adapted from [V 2-200:2012, 2.9]

**3.2.40 measured quantity value metadata**

metadata related to the measured quantity value

**3.2.41 measurement result relevant data**

data used during the process of constructing the measurement result

*Note:* Examples of measurement result relevant data include digital number or analogue value originating from a sensor or measuring instrument ID, in cases where it is part of the measurement result, see Annex C.

**3.2.42 measurement result relevant metadata**

metadata related to the construction of the measurement result

*Note:* Examples of measurement result relevant metadata include format of the digital number or analogue value originating from a sensor, format of the measured quantity value according to V 2:200:2012 or format of the measuring instrument ID, in cases where it is part of the measurement result.

**3.2.43 measurement result relevant information**

set of values of qualitative or quantitative variables relevant to the measurement result

*Note:* Measurement result relevant information includes measurement result relevant data and measurement result relevant metadata.

**3.2.44 metadata**

data about data or data elements, possibly including their data descriptions, and data about data ownership, access paths, access rights and data volatility

[ISO/IEC 2382:2015 Information technology – Vocabulary]

**3.2.45 mobile app**

computer program or software application designed to run on a mobile device such as a phone, tablet, or watch

[Cambridge Dictionary]

**3.2.46 non-interruptible cumulative measurement**

cumulative measuring process with no definite end that cannot be stopped and continued again by a user/operator without falsifying the result of the measurement

*Note 1:* Examples include: a) continuous totalising automatic weighing instrument, b) heat meter.

*Note 2:* See also interruptible cumulative measurement (3.2.26).

**3.2.47 protective interface**

legally relevant software module that handles all data flow to the legally relevant software modules(s) in order to prevent inadmissible influences

**3.2.48 remote verification**

set of procedures to support verification of an instrument during use, potentially without a person on site

**3.2.49 sealing**

means intended to protect the measuring instrument against any modification, readjustment, removal of parts or software, etc.

*Note:* This may be achieved by hardware, software or a combination of both.

adapted from [OIML V 1:2013, 2.20]

**3.2.50 securing**

means preventing unauthorised access to hardware or software

*Note:* This may be achieved by means of passwords.

adapted from [OIML V 1:2013, 2.21]

**3.2.51 significant defect**

incident that has an undesirable impact on the compliance of the measuring instrument or a fault

*Note:* Examples of significant defect include: a) deletion of the audit trail; b) inadmissible parameter changes; c) unauthorised updates d) accidental software changes due to physical effects.

**3.2.52 snapshot**

static representation of a dynamic module of legally relevant software at a specific point in time that can include 1) algorithm design (e.g. topology and weights of a neural network); 2) trail of evolution of dynamic parameters of module; 3) evolved parameters of the dynamic parts of the module

**3.2.53 software configuration management**

process to establish and maintain the integrity of the legally relevant software of a measuring instrument

*Note:* Configuration management as a discipline covers all aspects of legally relevant parts of the measuring instrument, whether software or hardware. However, this document only covers the software related requirements. Configuration management regarding hardware parts are to be given in the relevant Recommendation.

adapted from [ISO/IEC/IEEE 12207: 2017, 6.3.5]

**3.2.54 software examination**

technical operation that consists of determining one or more characteristics of the software according to the specific procedure (e.g. analysis of technical documentation or running the program under controlled conditions)

**3.2.55 software identification**

sequence of readable characters (e.g. version number, checksum) that represents the software or software module under consideration

*Note:* Software identification can be checked on an instrument whilst in use, see 6.2.1.

**3.2.56 software interface**

program code and dedicated data domain; receiving, filtering, or transmitting data between *software modules*

*Note 1:* A software interface is not necessarily legally relevant.

*Note 2:* A software interface is an interface between two or more software modules, used to exchange data and transmit commands.

[OIML V 1:2013, 6.03]

**3.2.57 software module**

software entity such as a program, subroutine, library, parameter or data set, and other objects including their *data domains* that may be in relationship with other entities

*Note:* The software of measuring instruments consists of one or more software modules.

**3.2.58 software protection**

protection of measuring instrument or component software or data domain by a hardware or software implemented seal with the intention of making an intervention impossible or evident

Examples:

- 1) A hardware seal on a measuring instrument's housing needs to be removed, damaged or broken to obtain access to change software.

- 2) A software seal in a measuring instrument records events, i.e. either a non-resettable counter is incremented each time an event occurs, see 3.2.20, or a data file, containing time stamped information, records the event, see 3.2.1.
- 3) The interface of a measuring instrument is physically sealed, so that accessing that interface can only be achieved by breaking, removing or damaging the seal.

*Note:* See 6.2.3.5.

adapted from [OIML V 1:2013, 6.04]

### 3.2.59 **software separation**

separation of the software in measuring instruments, which can be divided into legally relevant module(s) and legally non-relevant module(s)

*Note:* These module(s) communicate via a software interface.

adapted from [OIML V 1:2013, 6.02]

### 3.2.60 **source code**

computer program written in a form (programming language) that is legible and editable

*Note:* Source code is compiled or interpreted into executable code.

### 3.2.61 **storage device**

device used for storing measurement data that is necessary to construct the measurement result

adapted from [OIML V 1:2013, 6.07]

### 3.2.62 **test item**

property or function of a software module that may be subject to a test

*Note 1:* Test items are typically examined and tested and as part of remote verification procedures.

*Note 2:* Examples of potential test items include correctness of algorithms, software identity and software integrity.

### 3.2.63 **time stamp**

unique value, e.g. in seconds or a date and time string denoting the date and/or time at which a certain incident (e.g. measurement or event) occurred

### 3.2.64 **transmission of measurement data**

electronic transportation of measurement data via communication lines or other means to a receiver

**3.2.65 type (pattern) evaluation**

conformity assessment procedure on one or more specimens of an identified type (pattern) of measuring instruments which results in an evaluation report or a certificate

[OIML V 1:2013, 2.04]

**3.2.66 type-specific parameter**

legally relevant parameter with a value that depends on the type of instrument, component and/or module subject to legal control

*Note:* Type-specific parameters are part of the legally relevant software.

adapted from [OIML V 1:2013, 4.11]

**Example:**

Considering a measuring instrument intended for the dynamic measurement of liquids other than water, the range of kinematic viscosities of a turbine is a type-specific parameter, determined by the type evaluation of the turbine. All the manufactured turbines of the same type use the same range of viscosity.

**3.2.67 universal device**

device that is not constructed for a specific purpose, but that can be adapted to a legally relevant task by software

**3.2.68 user interface**

interface that enables information to be interchanged between the user/operator and the measuring instrument or its (hardware) components or (software) modules

*Note:* Typical examples of user interfaces are switches, keyboard, mouse, display, monitor, printer, touchscreen, software window on a screen including the software to generate it.

**3.2.69 verification**

provision of objective evidence that a given item fulfils specified requirements

[adapted from OIML V 2-200:2012, 2.44]

**3.2.70 verification of a measuring instrument**

conformity assessment procedure (other than type evaluation) which results in the affixing of a verification mark and/or issuing of a verification certificate

*Note:* See also OIML V 2-200:2012, 2.44.

[OIML V 1:2013, 2.09]



### 3.3 Abbreviations

CRC	Cyclic Redundancy Check
EUT	Equipment Under Test
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
IT	Information Technology
MPE	Maximum Permissible Error
OIML	International Organisation of Legal Metrology
PG	Project Group

## 4 Instructions for use of this Document in drafting OIML Recommendations

- 4.1** The provisions of this Document apply only to new OIML Recommendations and to OIML Recommendations under revision. OIML Project Groups (Technical Committees, Subcommittees) should use this guidance document to establish software-related requirements in addition to the other technical and metrological requirements of the applicable OIML Recommendation.
- 4.2** All referred documents are subject to revision, and the users of this Document are encouraged to investigate the possibility of applying the most recent editions of the referred documents.
- 4.3** It is the objective of this Document to provide the Project Groups responsible for drawing up OIML Recommendations with a set of requirements – partly with different (risk) levels – that are suitable to cover the demands of all kinds of measuring instruments and all areas of application. The Project Group shall determine which risk level is suitable, and how to incorporate the relevant portions of this Document into the OIML Recommendation being drafted. In Clause 5 some aid is given for performing this task.
- 4.4** PGs should define which influence is considered inadmissible for specific types of instruments.
- 4.5** PGs shall decide which measurement data is legally relevant and shall comply with the requirements, see Annex C. The manufacturer shall document the required metadata where necessary, see 7.1.2.
- 4.6** PGs should decide which parameters are relevant for a specific application.
- 4.7** PGs shall decide which metrological characteristics (at least legally relevant software, parameters and measurement data) shall comply with the requirements laid out in the following clauses.

## 5 Risk assessment

**5.1** This clause is intended as a guide to determine a set of risk levels to be generally applied for tests carried out on software-controlled measuring instruments. It is not intended as a classification with strict limits leading to special requirements, as in the case of an accuracy classification.

Moreover, this Document does not restrict Project Groups from providing risk assessments that differ from those resulting from the guidelines set forth in this Document. Different risk levels may be used in accordance with special limits prescribed in the relevant Recommendations.

**5.2** When selecting risk levels for a particular category of instruments and area of application (trade, direct selling to the public, health, law enforcement, etc.), the following aspects can be taken into account:

- a) risk of fraud:
  - the consequence and the social and societal impact of malfunction;
  - the value of the goods to be measured;
  - platform used (built-for-purpose or universal devices);
  - exposure to sources of potential fraud (unattended self-service device).
- b) required conformity:
  - the practical possibilities for the industry to comply with the prescribed level.
- c) required reliability:
  - environmental conditions;
  - the consequence and the social and societal impact of errors.
- d) motivation of the defrauder.
- e) the possibility to repeat a measurement or to interrupt it.

PGs should consider risk assessment standards when deciding risk levels, e.g. ISO/IEC 27005 [10].

Throughout the requirements clauses (see 6), various examples of acceptable technical solutions are given illustrating the basic level of protection against fraud, conformity, reliability, and type of measurement (marked with (I)). Where suitable, examples with enhanced counter measures are also presented that consider a raised risk level of the aspects described above (marked with (II)).

The examination level and risk level are linked. A deep analysis of the software shall be performed when a raised risk level is required in order to detect software deficiencies or security vulnerabilities. On the other hand, mechanical sealing (e.g. sealing of the communication port or the housing) should be considered when choosing the examination level.

## **6 Requirements for measuring instruments with respect to the software**

### **6.1 General**

The requirements are separated into general requirements (6.2), applicable to all kinds of measuring instruments, and requirements for specific configurations (6.3), additional requirements for technical features not applicable in all areas of legal application.

In the examples, where applicable, both normal and raised risk levels are shown. Notation in this Document is as follows:

- (I) Technical solution acceptable in case of normal risk level;
- (II) Technical solution acceptable in case of raised risk level (see 5).

### **6.2 General requirements**

At the time of publishing this Document, the general requirements represent the state of the art in information technology (IT). They are in principle applicable to all kinds of software-controlled measuring instruments and components of measuring instruments. They should be considered in all Recommendations. In contrast to these general requirements, the requirements specific for configurations (6.3) deal with technical features that are not common for some kinds of instruments or in some areas of application.

#### **6.2.1 Software identification**

Software modules of a measuring instrument or component shall be unambiguously identified. The software identification (see 3.2.55) linked to the software may consist of more than one part. But at least one part shall be dedicated to the legal purpose.

*Note:* The software identification is a legally relevant parameter.

The identification shall be displayed or printed by the measuring instrument:

- on command; or
- during operation; or
- at start-up for a measuring instrument that can be turned off and on again.

If a measuring instrument or component has neither display nor printer or if the instrument facilitates remote verification, the identification shall be sent via a communication interface, in order to be displayed/printed on another component or by the verification software.

As an exception, an imprint of the software identification on the instrument or component shall be an acceptable solution if it satisfies all of the following conditions:

- a) The user interface does not have any control capability to activate the indication of the software identification on the display, or the display does not technically allow the identification of the software to be shown (analog indicating device or electromechanical counter).
- b) The instrument or component does not have an interface to communicate the software identification.
- c) After production of the instrument or component a change of the software is not possible, or only possible if the hardware is also changed.

The software identification shall be correctly marked on the instrument or component concerned. The relevant Recommendation should allow or disallow this exception.

If the software is modified in any way, a new software identification is required.

Regardless of the form of the software identification it shall be accessible, to allow for it to be checked, at any time the instrument is in-service.

The software identification and the means of identification (e.g. software version, hash value, checksum, CRC) shall be stated in the certificate. Instructions on how to display or print the software identification shall be in the certificate.

*Note 1:* If measuring instruments in use need to conform to a certified type, software identification enables surveillance personnel and persons affected by the measurement to determine conformance of the measuring instrument.

*Note 3:* A software identification is a legally relevant parameter. When the software identification consists of more than one part, at least the part dedicated to the legal purpose constitutes a legally relevant parameter.

*Note 4:* A software separation (3.2.59), which includes an identification of a legally relevant part, may be considered depending on the structure of the measuring instrument or component. In this case, applicable requirements are given in 6.3.2.2.

**Examples:**

- 1) (I) The software contains a textual string or a number, unambiguously identifying the installed version. This string is transferred to the display of the instrument when a button is pressed, when the instrument is switched on, or cyclically controlled by a timer.

A version number has the following structure: A.Y.Z. Considering a flow computer; the letter A will represent the version of the core software that is counting pulses; the letter Y will represent the version of the conversion function (none, at 15 °C, at 20 °C); the letter Z will represent the language of the user interface.

- 2) (II) The software calculates a checksum of the executable code and presents the result as the identification instead of or in addition to the string in 1).

## 6.2.2 Correctness of algorithms and functions

The measuring algorithms and functions of a measuring instrument shall be appropriate and functionally correct for the given application and device type (accuracy of the algorithms, price calculation according to certain rules, rounding algorithms, etc.).

The measurement result (measured quantity value and measurement result relevant data required by specific Recommendations or by national legislation) shall be displayed or printed correctly.

It shall be possible to examine algorithms and functions either by metrological tests, software tests or software examination (as described in 7.3).

No hidden or undocumented functions or parameters shall exist.

*Note:* The requirement regarding hidden functions only applies to legal metrology.

## 6.2.3 Evidence and prevention of intervention

- 6.2.3.1 Software shall be protected against any changes, for example due to physical effects and intentional misuse, i.e. modification, loading or changes by swapping the memory device, unauthorised updates.

*Note:* Downloading software into the measuring instrument or component is allowed if the requirements for download are fulfilled, see 6.3.8.3 and 6.3.8.4.

During processing, measurement data shall be protected and secured.

*Note:* Protection of the measurement data can be achieved by ensuring that only legally relevant software can process them and that all interfaces are protected.

Software shall be protected in such a way that evidence of any intervention (e.g. software updates, parameters changes) shall be available. Mechanical sealing or software seals shall be used to protect measuring instruments or components.

*Note:* In case of a software implemented seal, see clause 6.2.6.1 regarding requirements on checking facilities and appropriate reactions.

In case of dynamic modules of legally relevant software with predefined parameters, these shall be considered as a part of the software and treated as such. This entails logging of all parameter changes in an audit trail (see 3.2.1).

If necessary for the purpose of verification, data containing evidence of an intervention shall be displayed or printed on command and, if applicable, transmitted to the verification software.

*Note 1:* If legally relevant software runs on a universal device such as a smartphone, it may not be possible to fully secure the software as required. Instead, additional external protection means (e.g. cryptographic signatures for transmitted or indicated measurement data) may be used to check correct behaviour of the software.

*Note 2:* If a legally relevant parameter is changed, a reverification might be required depending on national legislation. To allow for the possibility of parameter adaptations in dynamic modules of legally relevant software without reverification, the source of the parameter change (e.g. the learning facility) is logged in the audit trail, see 6.2.3.6.

The following examples 1) to 4) illustrate means of protecting software against intentional modification.

**Examples:**

- 1) (I) A measuring instrument consists of two components, one containing the main metrological functions incorporated in a housing that is sealed. The other component is a universal device with an operating system. Some functions such as the indication are located in the software of this device. To prevent swapping of the software on the universal device the transmission of measurement data between the component and the universal device is encrypted. The key for decryption is included in a program that is part of the legally relevant software of the universal device. Only this program knows the key and is able to read, decrypt and use the measurement data. Other programs cannot be used for this purpose as they cannot decrypt the measurement data (see also example in 6.3.2.2.2).
- 2) (I)/(II) The housing containing the memory devices is sealed or the memory device is sealed on the printed circuit board.
- 3) (II) The write-enable input of the device is inhibited by a switch that can be sealed. The circuit is designed in such a way that the write protection cannot be cancelled by a short-circuit of contacts.
- 4) (I) The software contains a neural network of fixed topology, but with flexible weights that change from time to time, to affect the measuring algorithm's behaviour.

A checksum over all weights in predefined order is used to identify the neural network weights, while a version number is used for the neural network overall structure and the rest of the software. The checksum is updated and logged in an audit trail, everytime that the parameters change.

6.2.3.2 All inputs from the user interface shall be handled by a protective interface. Any function that can be activated by the user interface shall:

- be clearly documented (see 7.1.2)
- not be able to influence the legally relevant characteristics of the instrument.

*Note:* The type evaluation authority decides whether all of these documented functions are acceptable.

*Example:*

(I)/(II) All inputs from the user interface are redirected to a protective interface that filters incoming commands. It only allows the commands to trigger the documented functions deemed acceptable by the type evaluation authority because they do not influence the legally relevant characteristics and discards all others. This module is part of the legally relevant software.

6.2.3.3 All inputs from communication interfaces shall be handled by a protective interface. Any function that can be activated through a communication interface shall:

- be clearly documented (see 7.1.2)
- not be able to influence the legally relevant characteristics of the instrument remotely such as through a remote verification procedure or a software download.

*Note:* The type evaluation authority decides whether all of these documented functions are acceptable.

6.2.3.4 Legally relevant parameters shall be secured and protected in such a way that evidence of an intervention shall be available. If necessary for the purpose of verification of a measuring instrument, displaying or printing and, if applicable, transmitting the current relevant parameter settings to the verification software shall be possible. The relevant Recommendation may require the setting of certain device-specific parameters to be available to the user. If that is so, the measuring instrument shall be fitted with a facility to automatically and non-erasably record any adjustment of the legally-relevant device-specific parameter, e.g. an audit trail, see 6.2.3.6.

*Note 1:* Type-specific parameters have identical values for all specimens of a type. They are fixed at type evaluation of the instrument.

*Note 2:* In case of a software implemented seal, see clause 6.2.6.1 regarding requirements on checking facilities and appropriate reactions.

*Note 3:* The audit trails are part of the legally relevant software, see 6.2.3.6.

*Example:*

(I)/(II) Device-specific parameters to be protected are stored in a non-volatile memory. The write-enable input of the memory is inhibited by a switch that is sealed. Refer to examples 6.2.3.5 1) to 3).

- 6.2.3.5 Software protection means shall comprise appropriate sealing by mechanical, software and/or cryptographic means, making an intervention impossible or evident.

*Note:* A cryptographic certificate may be used. The software is signed by a trustworthy institution with an electronic signature. The authenticity of the signed software can be verified by using the public key of the trustworthy institution and decrypting the signature of the certificate.

Examples:

- 1) (I) Electronic sealing. The legally relevant parameters of an instrument can be input and adjusted by a menu item. The software recognises each change and increments an event counter with each event of this kind. This event counter value can be indicated. The initial value of the event counter is marked durably on the instrument. If the indicated value differs from the registered one, the instrument is in an unverified state (equivalent to a broken seal).
- 2) (I)/(II) The software of a measuring instrument is constructed such (see example 6.2.3.1) that there is no way to modify the legally relevant parameters except via a switch protected menu. This switch is mechanically sealed in the inactive position, making modification of the legally relevant parameters impossible.  
  
To modify the legally relevant parameters, the switch needs to be activated, inevitably breaking the seal by doing so.
- 3) (II) The software of a measuring instrument is constructed such that there is no way to access the legally relevant parameters except by authorised persons. If a person wants to access the parameter menu item, that person needs to insert their smart card containing a personal identification number (PIN) as part of a cryptographic certificate. The software of the instrument is able to verify the authenticity of the personal identification number (PIN) by the certificate and allows the parameter menu item to be entered. The access and any parameter changes are recorded in an audit trail including the identity of the person (or at least of the smart card used).

- 6.2.3.6 Audit trails and event counters are part of the legally relevant software and shall be secured and protected as such. It shall not be possible to delete or inadmissibly change the data of the event counter or audit trails and it shall not be possible to exchange the audit trails or the value of the event counter when the software is updated. The audit trail shall contain at minimum the following information:

- time stamp of the event;
- in the case of a traced update, see 6.3.8.4.8;
- in the case of a parameter change:
  - Identification of the changed parameter;
  - The old and new value of the changed parameter.

If applicable, the source of the modification shall be recorded in the audit trail.

The audit trail or value of the event counter shall be displayed or printed on command and, if applicable, transmitted to the verification software. The certificate shall describe how the audit trail or the value of the event counter may be displayed or printed and specify if the audit trail or event counter is part of the remote verification procedure.

#### 6.2.4 Prevention of misuse

A measuring instrument shall be constructed in such a way that possibilities for unintentional, accidental, or intentional misuse are minimal. In the framework of this Document, this applies especially to the software.

*Note:* Software-controlled instruments are often complex in their functionality. The user needs good guidance for correct use and for achieving correct measurement results.

The following example 1) illustrates possible means of preventing unintentional or accidental misuse. Example 2) illustrates possible means of preventing unintentional, accidental and intentional misuse.

Examples:

- 1) (I) The user is guided by menus. The legally relevant functions are combined into one branch in this menu. If any measurement data might be lost by an action, the user is warned and requested to perform another action before the function is executed. See also 6.3.3.
- 2) (I) The measurement is started remotely by a smartphone app, which runs on an arbitrary device. The measuring instrument itself is fully secured and protected (physically and in software). It only allows one single command as input for starting a measurement via a protective interface. Once the measurement is completed the result is indicated on a display attached to the instrument. The result is also sent back to the smartphone for secondary indication.

Legally relevant software shall be secured against accidental or unintentional changes.

The presentation of the measurement results shall be unambiguous for all parties affected.

#### 6.2.5 Demands on the user

The software of a measuring instrument shall be designed in such a way that no unreasonable demands are required from the user to obtain a correct measurement result.

#### 6.2.6 Support of hardware features

##### 6.2.6.1 Detection of significant defects

The relevant Recommendation may require detection functions for significant defects and specify at what time and/or in which timeframe a check shall be carried out. In this case, the manufacturer of the instrument shall be required to design checking facilities into the software modules or hardware components or provide means by which the hardware components can be supported by the software modules of the instrument.

If software is involved in the detection of significant defects, it shall appropriately act upon any detected defect. For example, the relevant Recommendation may prescribe that the instrument or component is deactivated or an alarm and/or record in an error log is generated in case a significant defect is detected.

The documentation to be submitted for type evaluation shall contain a list of the significant defects that will be detected by the software, how it will act upon these defects and in case needed for understanding its operation, a description of the detecting algorithm, see 7.1.2.



Examples:

- 1) (I) On each start-up the legally relevant software calculates a checksum of the program code and legally relevant parameters. The nominal value of these checksums has been calculated in advance and stored in the instrument. If the calculated and stored values do not match, the legally relevant software stops execution.  
  
In case of a non-interruptible cumulative measurement, the checksum is calculated cyclically and controlled by a software timer. In case a failure is detected, the software displays an error message or switches on a failure indicator and records the time of the significant defect in an error log.
- 2) (II) On each start-up, the legally relevant software calculates a value produced by a cryptographic hash function of the program code and legally relevant parameters. The nominal value of the hash has been calculated in advance and stored in the instrument. If the calculated and stored values do not match, the program stops execution.  
  
In case of a non-interruptible cumulative measurement, the hash value is calculated cyclically and controlled by a software timer. In case a failure is detected, the software displays an error message or switches on a failure indicator and records the time of the significant defect in an error log.

#### 6.2.6.2 Durability protection

The relevant Recommendation may require detection functions for durability errors and specify at what time and/or in which timeframe a check shall be carried out.

*Note:* It is the manufacturer's choice to realise durability protection facilities addressed in OIML D 11:2013 [2] (5.1.3 (b) and 5.4) in software or hardware, or to allow hardware facilities to be supported by software. The relevant Recommendation may suggest appropriate solutions.

If software is involved in durability protection, it shall appropriately act upon any detected durability error. For example, the relevant Recommendation may prescribe that the instrument or component is deactivated or an alarm and/or record in an error log is generated in case durability is detected as being jeopardised.

The documentation to be submitted for type evaluation shall contain a list of the significant durability errors that will be detected by the software, how it will act upon these errors and, in case needed for understanding its operation, a description of the detecting algorithm, see 7.1.2

Example:

(I)/(II) Some kinds of measuring instruments require an adjustment after a prescribed time interval, in order to guarantee the durability of the measurement. The software gives a warning when the maintenance interval has elapsed and even stops measuring, if it has been exceeded for a certain time interval.

#### 6.2.6.3 Information for remote verification

If support of 6.2.6.1 or 6.2.6.2 is part of the remote verification procedure it shall be possible to transmit data containing information in this respect to the verification software.

### 6.2.7 Time stamps

The time stamp shall be in a consistent format, allowing for easy comparison of two different records and tracking progress over time.

If a measuring instrument uses time stamps, the instrument shall contain an internal clock which shall be used for the creation of the time stamp. Depending on the kind of instrument or on the field of application, setting the clock may be legally relevant and appropriate protection means shall be taken according to the risk level to be applied (see 6.2.3.4). Automatic setting of the time shall only be possible if legal time is used as a time base in an authenticated manner. If an internal clock is synchronized with legal time, the method of synchronization and traceability to legal time shall be described, see 7.1.2.

*Note 1:* The internal clock of a stand-alone measuring instrument may have a rather large uncertainty if no means are incorporated to synchronise this clock with legal time. Where the specific field of application requires high accuracy information concerning the exact time of the measurement, it may be necessary to improve the reliability of the internal clock using specific means.

*Note 2:* Where relevant, PGs may define requirements and test methods for internal clocks.

The use of time stamps shall be mandatory if audit trails are used.

**Example:**

(II) The reliability of the internal quartz-controlled clock device of the measuring instrument is enhanced by redundancy. A timer is incremented by the clock of the microcontroller that is derived from another quartz crystal. When the timer value reaches a preset value, e.g. 1 second, a specific flag of the microcontroller is set and an interrupt routine of the legally relevant software increments a second counter. The second counter is represented in the “date and time” format according to ISO 8601 [13]. At the end of e.g. one day the software reads the quartz-controlled clock device and calculates the difference in the seconds counted by the software. If the difference is within predefined limits, the software counter is reset and the procedure repeats; but if the difference exceeds the limits, the software initiates an appropriate error reaction.

## 6.3 Requirements for specific configurations

### 6.3.1 General

The requirements given in clause 6.3 are based on typical technical solutions in information technology, although they might not be common in all areas of legal applications. When following these requirements, technical solutions are possible that show the same degree of security and conformity to a type as instruments that are not software-controlled.

### 6.3.2 Specification and separation of legally relevant components and modules and requirements for interfaces

This requirement applies if the measuring instrument or component has interfaces for communicating with other devices, components or with other software modules besides the legally relevant software modules within a measuring instrument or component.

*Note:* With respect to the user interface, see 6.2.3.2.

Legally relevant software modules or hardware components of a measuring instrument shall not be inadmissibly influenced by another device or by other modules or components of the measuring instrument.

Recommendations may specify the software, hardware and data or part of the software, hardware and data that are legally relevant.

#### 6.3.2.1 Separation of components

##### 6.3.2.1.1 Components of a measuring instrument that perform legally relevant functions shall be identified, clearly defined and documented, see 7.1.2. They form the legally relevant hardware of the measuring instrument.

*Note 1:* The type evaluation authority decides whether the legally relevant hardware is complete and whether other components of the measuring instrument may be excluded from further evaluation.

*Note 2:* With respect to separation of software modules, see 6.3.2.2.

#### Examples:

- 1) (I)/(II) An electricity meter with a local display is equipped with a protective optical interface for connecting an electronic device to read out the measurement result. The meter stores all measurement results and keeps the results available to be read out for a sufficient time span. In this system, only the electricity meter is the legally relevant instrument. Other legally non-relevant devices can be connected to the protective interface that complies with 6.3.2.1.2. Securing of the data transmission itself (see 6.3.5) is not required.
- 2) (I)/(II) A measuring instrument consists of the following components:
  - a digital sensor that calculates the weight or volume;
  - a universal device that calculates the price;
  - a printer that prints out the measurement result and the price to pay.

All components are connected by a local area network. In this case the digital sensor, the universal device and the printer are legally relevant components and are optionally connected to a merchandise system that is not legally relevant. The legally relevant components fulfil requirement 6.3.2.1.2 and – because of the transmission via the network – also the requirements contained in 6.3.5.

- 6.3.2.1.2** A legally relevant software-controlled component shall communicate with other components or devices through a protective interface. It shall not be possible to inadmissibly influence the legally relevant software, parameters or measurement data through these interfaces, see also 6.3.6.2. It shall be demonstrated that legally relevant software, parameters and measurement data of components that are legally relevant cannot be inadmissibly influenced by commands received via the protective interface.

This implies that there is an unambiguous assignment of each command to all initiated functions or data changes in the component.

*Note:* If “legally relevant” components interact with other “legally relevant” components, refer to 6.3.5.

**Examples:**

- 1) (I)/(II) The software of the electricity meter (see example (1) of 6.3.2.1.1 above) is able to receive commands for selecting the measurement results required. It sends the measurement result (including additional measurement result relevant data – e.g. time stamp, unit) back to the requesting device. The software only accepts commands for the selection of valid allowed quantities and discards any other command, sending back only an error message. Securing means for the contents of the dataset are not required, as the transmitted dataset is not subject to legal control.
- 2) (I)/(II) Inside the housing that is sealed there is a switch that defines the operating mode of the electricity meter: one switch setting indicates the secured mode and the other the free mode (securing means other than a mechanical seal are possible; see examples 6.2.3.1 and 6.2.3.5). When interpreting received commands, the software checks the position of the switch: in the free mode, the command set that the software accepts is extended compared to the secured mode (e.g. it is possible to adjust the calibration factor by a command that is discarded in the secured mode).

- 6.3.2.1.3** PGs may decide that legally relevant components shall be protected against exchange.

If software seals are used to prevent components from being exchanged and pairing parameters are part of the seal, then these pairing parameters are legally relevant and shall be secured and protected in such a way that evidence of an intervention is available, see 6.2.3.4.

*Note:* Pairing parameters could also include network or internet (IP) address.

**Examples:**

- 1) (I) When a new component is connected to an existing measuring instrument via ethernet, a secret 32-bit binary pairing key is manually entered into the component and into the measuring instrument. As additional pairing parameters, the network address of the respective communication partner is also manually set. Whenever one side or the other exchanges data with the communication partner under the specified network address, they symmetrically encrypt their communication using RC4 with the secret pairing key.
- 2) (II) When a new component is connected to an existing measuring instrument via ethernet, both sides exchange X.509 cryptographic certificates signed by the manufacturer and log the exchange in an audit trail. Whenever they exchange data, they sign them using an ECC-based signature using the secret key corresponding to the certificate. The origin of the signed data is verified by the receiver using the available certificate. If the signature of the sender cannot be verified, the receiver displays an error message and prevents further measurements.

- 6.3.2.1.4** PGs may decide that legally relevant components shall check the authenticity, integrity and/or availability of another software-controlled component. In case the authenticity and/or integrity check fails, or the other component is not available, the checking component shall appropriately act upon this. See 6.2.6.1.

*Note:* In the case of simple recipient printers it could be that only availability needs to be checked.

- 6.3.2.1.5** If a component is shared by multiple components, e.g. one display for multiple sensors, then all the components that share another component shall be unambiguously identified.

*Note:* This requirement does not impose any restrictions on the manner of identification.

- 6.3.2.1.6** If some components of a measuring instrument are not physically connected and therefore present in the same location, it might be difficult to establish if an indicated result actually stems from and is indicated by legally relevant software. In case the completeness of the measuring instrument cannot be visually checked (e.g. wireless or network-connected components), non-legally relevant software modules shall be prevented from calculation/presenting/spoofing the measurement result.

Example:

(I/II) A measuring instrument consists of two components, one containing the main metrological functions incorporated in a housing that is sealed. The other component is a universal device with an operating system. Some functions such as the indication are located in the software of this device. To ensure that only the legally relevant software on the universal device can further process the measurement data the measurement data is encrypted. The key for decryption is included in a program that is part of the legally relevant software of the universal device. Only this program knows the key and is able to read, decrypt and use the measurement data. Other programs cannot be used for this purpose as they cannot decrypt the measurement data (see also example in 6.3.2.2.2).

- 6.3.2.1.7** PGs may decide that functionalities in certain components shall be restricted, for example the functionality of apps on smartphones or when cloud storage devices are used.

In case legally relevant components with limited functionality and limited securing/protection capabilities are applied (e.g. if a legally relevant operating system on a component cannot be configured according to 6.3.6), they shall have limited access to the measurement data, i.e. they shall use the measurement data without modification or further processing.

- The measurement data shall be prepared for transmission or storage for further processing by a component that can be fully secured and protected. This component ensures that the data are complete and protected.
- The measurement data shall be received or retrieved for further processing by a component that can be fully secured and protected. This component ensures that the data are complete and shall check their integrity. The component also ensures that the measurement result is printed or indicated in case of a dispute.

Example:

(I) The measurement is started remotely by a smartphone app, which runs on a dedicated device belonging to the owner of the measuring instrument. The instrument itself is fully secured and protected (physically and in software) and only allows one single command as input for starting a measurement via a protective interface. Once the measurement is completed the result is cryptographically signed and sent back to the smartphone for indication as clear text accompanied by a two-dimensional bar code that contains measurement result and cryptographic signature. In case of doubt, the correct indication of the result can be checked by all parties by validating the signature contained in the two-dimensional bar code, see also 6.3.3. The signed measurement result can be uploaded to a secured and protected webserver which checks the signature and then indicates the result.

**6.3.2.1.8** PGs may decide that certain components shall be connected and available on site, for example a display or a printer.

Example:

(I/II) In the case an indication of a result is mandatory, a display is connected and available with the measuring instrument.

## 6.3.2.2 Separation of software modules

**6.3.2.2.1** All software modules (programs, subroutines, objects, operating system parts etc.), that perform legally relevant functions or that process legally relevant measurement data, form the legally relevant software of a measuring instrument or component. The conformity requirement applies to this software and it shall be made identifiable as described in 6.2.1.

If the separation of the software is not possible or needed, the software shall be legally relevant as a whole.

Software separation takes either place in the complete measuring instrument or in a specified component.

- For separation of components, see 6.3.2.1.
- For communication between components, see 6.3.5.

*Note:* If one or more dynamic modules of legally relevant software are used in combination with software separation, 6.2.3.1 needs to be observed to ensure that any parameter changes in these software modules are traceable.

Example:

(I) A measuring instrument consists of several digital sensors connected to a personal computer that displays the measurement result. The legally relevant software on the personal computer is separated from the legally non-relevant software by compiling all procedures realising legally relevant functions (including presentation of results) into a dynamically linkable library. This library contains all legally relevant functions like functions receiving the measurement data from the digital sensors, calculating the measurement result, and displaying it in a software window. One or several legally non-relevant applications may call functions in this library.

**6.3.2.2.2** All legally relevant software modules shall communicate with other modules through a protective interface. It shall be demonstrated that the functions and data of modules that are legally relevant cannot be inadmissibly influenced by commands received via the protective interface. The legally relevant software modules and the protective interface shall be clearly documented, see 7.1.2. All legally relevant functions and data domains of the software shall be described to enable a type evaluation authority to decide on correct software separation.

The protective interface consists of program code and dedicated data domains. Defined coded commands or data are exchanged between the software modules by storing to the dedicated data domain by one part of the protective interface and reading from it by another part of the protective interface. Writing and reading code is part of the protective interface.

Measurement data shall not be made available to legally non-relevant modules prior to primary indication. Furthermore, PGs may decide that no secondary indication is allowed for certain scenarios.

*Note 1:* Protection from interruptions (delayed execution or blocking by other processes) is addressed in 6.3.2.2.4.

*Note 2:* Software modules can be installed in a measuring instrument or in a component. With respect to separation of components, see 6.3.2.1.

Example:

(I) In examples 6.3.2.2.1 and 6.3.2.2.3, the legally non-relevant application controls the start of the legally relevant procedures in the library via a protective interface. Omitting a call of these procedures would of course inhibit the legally relevant function of the system. Therefore, the following provisions have been made in the example system: The digital sensors send the measurement data in encrypted form. The key for decryption is hidden in the library. Only the procedures in the library know the key and are able to read, decrypt measurement data, and display measurement results. Only after indication of the measurement results does the library allow other legally non-relevant modules to read the result.

**6.3.2.2.3** There shall be an unambiguous assignment of each command to all initiated functions or data changes in the legally relevant software. Functions that are triggered through the protective software interface shall be declared and documented, see 7.1.2. Only documented functions shall be activated through the protective software interface.

Examples:

- 1) (I) In the example described in 6.3.2.2.1 the protective software interface consists of the procedures in the library and their parameters and return values. The interface cannot be circumvented e.g. by pointers to internal data. The number and kind of procedures, parameters, and return values is fixed at compile time.
- 2) (II) Legally relevant and legally non-relevant software modules run in separate virtual machines on a universal device. Both machines are configured in such a way that any communication between both software modules can only be done via the defined protective software interface. The setup of the virtual machines, including the method of communication between both, is part of the legally relevant software. The operating system ensures that the configuration cannot be modified. The operating system configuration itself is protected by a sealed administrator password. Therefore, changes to the setup of the virtual machines cannot happen without breaking a seal.

- 6.3.2.2.4** Where the legally relevant software has been separated from the non-relevant software, the legally relevant software shall have priority using the resources over non-relevant software. The legally relevant process shall not be inadmissibly interrupted by legally non-relevant software. The measurement process (realised by the legally relevant software) shall not be delayed or blocked by other processes.

Examples:

- 1) (I) A priority level is assigned to the legally relevant function which is higher than for normal processes and which cannot be decreased by a user/operator of the measuring instrument.
- 2) (I) The software of an electronic electricity meter reads measurement data from an analog-digital converter (ADC). For the correct calculation of the measurement result the delay between the “data ready” signal from the ADC to finishing buffering of the measurement data is crucial. The measurement data are read by an interrupt routine initiated by the “data ready” signal. The instrument is able to communicate via an interface with other electronic devices in parallel served by another interrupt routine (legally non-relevant communication). The priority of the interrupt routine for processing the raw values is higher than that of the communication routine.
- 3) (II) Legally relevant and legally non-relevant software run in separate virtual machines on a universal device. The configuration of the operating system ensures that the virtual machine on which the legally relevant software runs always has sufficient system resources available for the legally relevant processes.

- 6.3.2.2.5** When dynamic modules of legally relevant software have facilities for continuous learning that allow dynamic parameter changes during use, the manufacturer shall clarify the facilities and its priorities to the whole legally relevant software, especially in reference to the measuring functions.

Where relevant, PGs may specify the requirement that the measuring functions shall not be inhibited/affected by the continuous learning process.

The software documentation shall contain the description of the prioritization of using all legally relevant parts including dynamic modules of legally relevant software, see 7.1.2.

### 6.3.3 Shared indications

A display or printout may be employed to present both information from the legally relevant software and other information. The contents and layout are specific to the kind of instrument and field of application and shall be defined in the relevant Recommendation. If a display or printout is used both for legally relevant and legally non-relevant outputs, the legally relevant information shall always be readable, and clearly distinguishable from other information.

Examples:

- 1) (I) In the measuring instrument described in the examples 6.3.2.2.1 to 6.3.2.2.4, the measurement results are displayed in a separate software window. The means described in 6.3.2.2.4 guarantee that the legally relevant software can read and display the measurement results before such data are made available to other legally non-relevant software modules. The instrument has an operating system with a multiple windows user interface. The window displaying the legally relevant data is generated and controlled by procedures in the legally relevant dynamically linkable library (see 6.3.2.2). During measurement, these procedures check cyclically that the relevant



- window is still on top of all the other open windows; if not, the procedures place it on top.
- 2) (II) In the measuring instrument described in the examples 6.3.2.2.1 to 6.3.2.2.4 the measurement application runs in kiosk mode. The entire display is controlled by the legally relevant software. Legally non-relevant data is presented in a special part of the display marked as legally non-relevant.
  - 3) (II) A smartphone app on a device belonging to the measuring instrument is used to indicate measurement results calculated on a separate component. Since the smartphone is also used for other legally non-relevant purposes, the operating system of the smartphone is configured according to clause 6.3.6. Whenever the legally relevant app is running, the user is informed by the app accordingly. To ensure that the measurement result can always be distinguished from legally non-relevant information, legally relevant measurement data is only made available to legally non-relevant apps after primary indication.

If increased protection against fraud is necessary (II), a printout as an indication alone may not be suitable and additional precautions in the form of hardware and/or software shall be considered. If so, a component shall exist with increased securing means that is able to display the measurement results.

Where measuring instruments or systems incorporate or are dependent upon dynamic modules of legally relevant software, this information shall be indicated and made available to any parties interested in the measurement result(s) produced by that measuring instrument or system.

Where a measurement result is the product of a measurement process that incorporates or is dependent upon dynamic modules of legally relevant software, the indication of the measurement result shall include information regarding the use of those modules in the measurement process. This may be achieved by the use of a short statement, clearly understood markings, symbols or other indications.

#### 6.3.4 **Storage of data**

##### 6.3.4.1 **General**

If measurement data are stored for legal purposes the requirements of 6.3.4.2 to 6.3.4.4 shall apply.

PGs may decide upon appropriate storage conditions for different applications.

Requirements regarding storage of data also apply to software identification, log files, results of diagnostics, result of remote verification, etc.

*Note 1:* PGs may define additional data which needs to be stored.

*Note 2:* Redundant entries do not count toward the total stored count in the audit trail.

##### 6.3.4.2 **Completeness of stored data**

The stored measurement data shall include all relevant data necessary for future legally relevant use.

Where measurement data is produced as a result of algorithms of dynamic modules of legally relevant software, the measurement data shall be marked or indicated as such. Such markings or indications and associated data shall form part of the legally relevant measurement data. PGs shall decide which measurement data (e.g. measurement result relevant data necessary to construct the measurement result) shall be stored.

**Example:**

- (I)/(II) A stored dataset of the measurement result includes the following entries:
- measured value including unit;
  - time stamp of measurement (see 6.2.7);
  - place of measurement or identification of the measuring instrument that was used for the measurement;
  - unambiguous identification of the measurement, e.g. consecutive numbers enabling assignment to values printed on an invoice;
  - a mark showing that the result originates from a dynamic module of legally relevant software if applicable.

**6.3.4.3 Protection of stored data**

The stored measurement data shall be protected by appropriate means to guarantee authenticity and integrity. The software that displays or further processes the measurement data shall check the authenticity and integrity of the data after having read them from the storage. If an irregularity is detected an appropriate response shall be required, for example the data shall be discarded or marked unusable.

The storage component shall have sufficient permanency to ensure that the stored measurement data are not corrupted under normal storage conditions.

Software modules that prepare data for storing, or that check data after reading are considered part of the legally relevant software.

*Note 1:* PGs may specify appropriate reactions to detected irregularities in stored data.

*Note 2:* It is appropriate to require a raised risk level when considering a freely accessible storage.

Raised risk levels might require the application of cryptographic methods. If appropriate, means shall be provided whereby cryptographic keys can only be input or read if a seal is broken. Example (1) applies to local storage and Example (2) applies to freely accessible storage. Example (3) addresses storage of measurement results in the cloud.

**Examples:**

- 1) (I) The program of the storing device calculates a CRC32 [11] of the dataset and appends it to the dataset. It uses a secret initial value for this calculation instead of the value given in the standard [11]. This initial value is employed as a key and stored as a constant in the program code. The reading program has also stored this initial value in its program code. Before using the dataset, the reading program calculates the checksum and compares it with that stored in the dataset. If both values match, the dataset is not falsified. Otherwise, the program assumes falsification and discards the dataset.
- 2) (II) The storing program that is part of the legally relevant software generates an electronic signature for the stored dataset. It is appended to the stored dataset. The private and public keys used for signing are generated in a hardware security module which protects the private key against manipulation or reading and exports the public key. The reading program verifies the signature with the public key to check the authenticity and integrity of the dataset. To prove the origin of the dataset the reading program needs to know whether the public key really belongs to the storing program. Therefore, the public key is presented on the display of the measuring instrument and can be registered once, e.g. together with the serial number of the instrument when it is verified in the field.

- 3) (II) Each dataset is stored in the cloud and protected by means of a cryptographic signature calculated by the Elliptic Curve Digital Signature Algorithm (ECDSA) with a key length of 256 bit. The private key used for signing is protected as in example 2). To ensure that no data are lost, each dataset includes a consecutive (paging) number whose current value is kept as a reference within the instrument. The measuring instrument periodically checks the completeness of the stored measurement datasets by randomly performing signature checks on previously exported datasets. A service level agreement between user and cloud service provider ensures that all datasets are available for inspection or verification purposes. Nevertheless, should one or more datasets be detected as missing, the measuring instrument notifies user and customer that data are lost. For individual datasets, the reading program always verifies the signature before indicating it.

#### 6.3.4.4 Automatic storing

- 6.3.4.4.1** When, considering the application, data storage is required, measurement data shall be stored automatically. A checking facility shall regularly check the availability of the storage and in the case the storage device is not available no measurement shall be possible, see 6.2.6.1.

There shall be sufficient memory storage for the intended application. PGs shall decide which action needs to be taken if the memory limit is reached (e.g. disabling further measurements).

If automatic storage is required, no measurement shall be possible if the storage device is not available.

When the data necessary for the calculation of the measurement result are relevant for legal purposes, all measurement result relevant data included in the calculation shall be automatically stored with the final value.

*Note 1:* In the case of cumulative measurements, it may happen that the same data domain (program variable) is used repeatedly. In that case, storage capacity may not be legally relevant.

*Note 2:* Stored data does not need to be physically localised in one storage unit, as long as all requirements are met.

Example:

(I)/(II) The program of the measuring instrument stores all datasets in a cloud. In case no communication connection to the cloud can be established, the instrument temporarily buffers new datasets until the cloud can be reached again and datasets are exported in first-in-first-out order. If the local buffer reaches its limit, further measurements are disabled.

- 6.3.4.4.2** Measurement data stored in a component to construct the measurement result can be deleted if the next module or component state a proper completion of expected actions engaged.

The measurement result may be deleted if

- the transaction is settled, or
- these data are printed by a printing device subject to legal control.

*Note:* Other general national regulations (e.g. for tax purposes) may contain strict limitations for the deletion of stored measurement data or results. PGs may define alternative conditions for data deletion.

### 6.3.5 Data transmission

#### 6.3.5.1 General

If measurement data are transmitted before they are used for legal purposes the requirements of 6.3.5.2 to 6.3.5.4 shall apply.

Requirements regarding data transmission also apply to software identification, log files, results of diagnostics, data transfer during remote verification, etc.

#### 6.3.5.2 Completeness of transmitted data

The transmitted measurement data shall include all data necessary for future legally relevant use.

Where measurement data is produced as a result of algorithms of dynamic modules of legally relevant software, the measurement data shall be marked or indicated as such. Such markings or indications and associated data shall form part of the legally relevant measurement data. PGs shall decide which measurement data (e.g. measurement result relevant data necessary to construct the measurement result) shall be transmitted.

Example:

(I)/(II) A transmitted dataset for the measurement result includes the following entries:

- measured value including unit;
- time stamp of measurement (see 6.2.7);
- place of measurement or identification of the measuring instrument that was used for the measurement;
- unambiguous identification of the measurement, e.g. consecutive numbers enabling assignment to values printed on an invoice;
- a mark showing that the result originates from a dynamic module of legally relevant software if applicable..

#### 6.3.5.3 Protection of transmitted data

The transmitted data shall be protected by software means to guarantee authenticity and integrity.. The software that displays or further processes the measurement data shall check authenticity and integrity of the data received from a transmission channel. If an irregularity is detected an appropriate reaction shall be required, for example the data shall be discarded or marked unusable.

Software modules that prepare measurement data for sending, or that check measurement data after receiving, are considered part of the legally relevant software.

*Note:* It is appropriate to require a raised risk level when considering an open network.

Raised risk levels might require application of cryptographic methods. Means shall be provided whereby cryptographic keys used by these methods can only be input or read if a seal is broken.

**Examples:**

- 1) (I) The legally relevant software of the sending device calculates a CRC32 [11] of the dataset. It is appended to the dataset. It uses a secret initial value for this calculation instead of the value given in the standard [11]. This initial value is employed as a key and stored as a constant in the program code. The legally relevant software of the receiving device has also stored this initial value in its program code. Before using the dataset, the program calculates the checksum and compares it with that stored in the dataset. If both values match, the dataset is not falsified. Otherwise, the program assumes falsification and discards the dataset.
- 2) (II) The legally relevant software of the sending device generates an electronic signature for the transmitted dataset. It is appended to the transmitted dataset. The private and public keys used for signing are generated in a hardware security module which protects the private key against manipulation or reading and exports the public key. The legally relevant software of the receiving device verifies the signature with the public key to check authenticity and integrity of the dataset. To prove the origin of the dataset the receiving program needs to know whether the public key really belongs to the transmitting program. Therefore, the public key is presented on the display of the measuring instrument and can be registered once, e.g. together with the serial number of the instrument when it is verified in the field.

**6.3.5.4 Transmission delay or interruption**

The measurement shall not be inadmissibly influenced by a transmission delay or interruption. If network services become unavailable or very slow, no measurement data shall be lost. It may be necessary to stop the measurement process to avoid the loss of measurement data. PGs shall decide upon appropriate requirements and mechanisms intended to preserve measurement data (e.g. disabling further measurements) where transmission interruptions are possible in the relevant application(s).

*Note 1:* Consideration should be given to distinguish between static and dynamic measurements.

*Note 2:* Depending on the area of application, and for cases where measurements are easily repeatable, a loss of transmitted data may be acceptable.

**Example:**

(I)/(II) The sending instrument or component waits until the receiver has sent an affirmation of correct receipt of the dataset. The sending instrument or component keeps the dataset in a buffer until this affirmation has been received. The buffer has a capacity for more than one dataset, organised as a FIFO (First-in-first-out) queue.

**6.3.6 Compatibility of operating systems and hardware****6.3.6.1 General**

If an operating system is part of the measuring instrument, requirements according to 6.3.6.2 to 6.3.6.7 shall be met.

Each of the following operating system requirements shall be met by measures on application level, operating system level or a combination of both. For example, the protective interface may be implemented within the legally relevant application, the operating system, the physical layer, etc.

### 6.3.6.2 Hardware interfaces

Hardware interfaces not equipped with a protective interface shall not be able to inadmissibly influence the legally relevant software, parameters or measurement data.

Examples:

- 1) (I) A legally relevant software module routinely checks all open physical interfaces for incoming traffic. In the case of inadmissible input, it inhibits measurements.
- 2) (I) A legally relevant software module interprets all commands reaching the legally relevant software and discards the inadmissible ones.
- 3) (II) All open interfaces are physically protected or disabled by the operating system.

### 6.3.6.3 Boot process

**6.3.6.3.1** If a secure boot process is needed to ensure protection of the legally relevant software, the requirements of 6.3.6.3.2 to 6.3.6.3.5 shall apply.

**6.3.6.3.2** The boot process shall ensure integrity and authenticity of the legally relevant software.

**6.3.6.3.3** If a chain of trust is established over the individual steps of the boot process to ensure 6.3.6.3.2, the processing of the chain of trust may be interrupted, as long as its integrity is preserved.

**6.3.6.3.4** The boot configuration shall be secured and protected.

Examples:

- 1) (I) The sealed housing of the measuring instrument together with protection of all open interfaces ensures that the boot configuration can only be modified after a seal has been broken.
- 2) (II) A TPM (trusted platform module) verifies the signature of the boot loader, the boot loader then verifies the operating system, which in turn verifies and starts the legally relevant application.

**6.3.6.3.5** Booting via open interfaces shall be prohibited.

### 6.3.6.4 System resources

The combination of the legally relevant software and the operating system shall ensure that there are enough resources for the operation of the legally relevant application.

Examples:

- 1) (I) The legally relevant application ensures that it has all the resources it requires.
- 2) (II) The minimum number of operating system parts is utilized to ensure the measurement process can be executed.

#### 6.3.6.5 **Protection during use**

**6.3.6.5.1** The operation of software that is not legally relevant shall not inadmissibly influence the legally relevant application.

**6.3.6.5.2** The combination of the legally relevant software and the operating system shall ensure that the legally relevant indication is distinguishable from other information.

**6.3.6.5.3** The access control feature of the operating system shall be configured in such way that the intended use cannot be inadmissibly influenced.

**6.3.6.5.4** The administration tasks of the legally relevant software shall be protected.

*Note:* The term “administration task” addresses all reconfigurations and updates of the operating system.

**Examples:**

- 1) (I) All legally relevant files are write-protected and the access permissions are routinely checked by the legally relevant software. Modifications of the permissions are logged in an audit trail.
- 2) (II) The legally non-relevant software runs in a virtually separated environment.

#### 6.3.6.6 **Communication with the legally relevant software**

Communication with the legally relevant software shall be made via protective interfaces.

It shall be demonstrated that the legally relevant software, parameters, and data of components that are legally relevant cannot be inadmissibly influenced by commands received via the protective interface, see also clause 6.3.2.2.2.

**Examples:**

- 1) (I) A legally relevant software module interprets all commands reaching the legally relevant software and discards the inadmissible ones.
- 2) (II) All open interfaces are physically protected or disabled by means of the operating system.

#### 6.3.6.7 **Identification and traceability**

**6.3.6.7.1** The configuration of the operating system shall be identifiable. The identifier shall be displayed on command or during operation and, if applicable, transmitted to the verification software by the measuring instrument.

The following examples 1) and 2) illustrate means of identifying the operating system configuration.

Examples:

- 1) (I)/(II) On a UNIX-type operating system, the configuration consists of legally relevant:
  - kernel modules
  - list of installed packages
  - libraries
  - accounts and user privileges
  - passwords
  - configuration files
  - file read/write/execute permissions
  - access to interfaces

All of the above is identified by means of a checksum.

- 2) (I)/(II) On a Windows operating system, the configuration consists of legally relevant:
  - kernel modules
  - list of installed packages
  - libraries
  - accounts and user privileges
  - passwords
  - configuration files
  - file read/write permissions
  - registry keys
  - access to interfaces

Each of the above is identified by means of a checksum.

**6.3.6.7.2** Legally relevant configuration settings of the operating system shall be protected, i.e. changes to the legally relevant configuration shall be traceable.

*Note 1:* Replacing one legally relevant operating system part with a different one, i.e. by a newer version, is considered a modification of the configuration.

*Note 2:* This implies that legally relevant operating system parts can only be changed by means of a verified update (see 6.3.8.3) or by means of a traced update (see 6.3.8.4) if an audit trail is used.

Example:

(I)/(II) All changes to the operating system configuration are logged in an audit trail. Each entry of the audit trail contains a time stamp of the modification as well as the identifier of the new configuration. The module in charge of maintaining the audit trail and protecting it against modification serves as a trust anchor and is not updated itself, see 6.3.8.4.4.

### 6.3.6.8 Suitable environment

The manufacturer shall identify the hardware and software environment that is suitable. Minimum resources and a suitable configuration management (e.g. processor, memory, specific communication, version of operating system, configuration management of



dynamic modules of legally relevant software, etc.) necessary to guarantee correct functioning of the legally relevant software shall be declared by the manufacturer and stated in the certificate.

#### 6.3.6.9 Constraints for operation

Technical means shall be provided in the legally relevant software to prevent operation, if the minimum resources or a suitable configuration are not met. The system shall be operated only in the environment specified by the manufacturer for its correct functioning.

Fixing the hardware, operating system, or system configuration of a universal device or even excluding the usage of an off-the-shelf universal device shall be considered in the following cases:

- if high conformity is required;
- if cryptographic algorithms or keys need to be implemented (see 6.3.4 and 6.3.5).

**Note:** The manufacturer shall identify and declare the impact of dynamic modules of legally relevant software (modules/parts/algorithms etc.) and it shall be stated in the certificate (see 6.3.2.2.5).

#### 6.3.7 Conformity of manufactured devices to the certified type

The manufacturer shall produce measuring instruments, components and legally relevant software that conform to the certified type and the documentation submitted.

*Note 1:* In the case of dynamic modules of legally relevant software, this implies that the documentation submitted describes a means to validate the conformity of devices in use even in presence of dynamic parameter changes, see 7.1.2.

*Note 2:* OIML D 34:2019 [12] interprets certification as consisting of type evaluation and type approval. Therefore, the term “certified type” used here corresponds to the term “approved type” in D 34.

#### 6.3.8 Maintenance and reconfiguration

##### 6.3.8.1 General

Updating the legally relevant software of a measuring instrument in use should be considered as

- a modification of the measuring instrument, when exchanging the software with another certified version, or
- a repair of the measuring instrument, when re-installing the same version.

A measuring instrument which has been modified or repaired while in service may require initial or subsequent verification, dependent on national regulations.

Software which does not realise legally relevant functions of the measuring instrument does not require verification after being updated.

##### 6.3.8.2 Applicability of update requirements

Only versions of the legally relevant software that conform to the certified type are allowed for use (see 6.3.7). They shall be stated in the certificate. Applicability of the following requirements depends on the kind of instrument and is to be worked out in the relevant Recommendation. The following options 6.3.8.3 and 6.3.8.4 are alternatives. In the case that device-specific parameters (especially calibration parameters) are concerned, only a verified update should be done.

This issue concerns verification of a measuring instrument in the field. Refer to clause 8 for additional constraints.

#### 6.3.8.3 **Verified update**

Verified Update is the procedure of changing software in a verified device or component after which the subsequent verification is necessary. The software to be updated may be loaded locally, i.e. directly on the measuring instrument, or remotely via a network. Loading and installation may be two different steps (as shown in Fig. 1) or combined into one, depending on the needs of the technical solution. A protection measure (i.e. physical or electronic seal that must be broken for the update to take effect) provides evidence of an intervention. A person should be on the installation site of the measuring instrument to check that the updated software has been installed successfully. After the update of the legally relevant software of a measuring instrument (exchange with another certified version or re-installation) the measuring instrument should not be employed for legal purposes before a verification of the measuring instrument as described in clause 8 has been performed and the securing means have been renewed and the protection means have been renewed or reactivated (if not otherwise stated in the relevant Recommendation or in the certificate).

#### 6.3.8.4 **Traced update**

##### 6.3.8.4.1

Traced update is the procedure of changing software in a verified instrument or component after which a subsequent verification is not necessary. This means the traced update shall not affect existing parameters. The software to be updated may be loaded locally, i.e. directly on the measuring instrument, or remotely via a network. The software update is recorded in an audit trail (see 3.2.1). The procedure of a traced update comprises several steps: loading, integrity checking, checking of the origin (authentication), installation, logging and activation. The software shall be implemented in the instrument according to the requirements for Traced update (6.3.8.4.2 to 6.3.8.4.9), if it is in compliance with the relevant Recommendation.

*Note:* PGs may specify procedures to test and evaluate traced updates to provide evidence that they do not affect the legally relevant parameters of the measuring instrument, and otherwise comply with all relevant requirements for traced updates.

##### 6.3.8.4.2

Depending on the needs and on national legislation it may be necessary for the user or owner of the measuring instrument to give their consent to a traced update. If so, the measuring instrument shall have a feature for the user or owner to express their consent prior to an update, e.g. by means of a push button. It shall be possible to enable and disable the feature, e.g. by a switch that can be sealed or by a secured and protected parameter. If the feature is enabled, each traced update needs to be initiated by the user or owner. If the user or owner denies consent, the update procedure should not start at all. If the feature is disabled, no activity by the user or owner is necessary to perform a traced update.

##### 6.3.8.4.3

After initiation of the update procedure, a traced update of software shall run automatically. If some of the securing or protection measures of the instrument are turned off to enable updating, they shall be turned on again immediately after update, independent of the result of the update process.

*Note:* Triggering of the traced update process may require intervention/manual actions by the user of the measuring instrument, see 6.3.8.4.2.

- 6.3.8.4.4** During a Traced update, any existing protection measures, e.g. audit trail information and event counter values, shall be retained.

Example:

(I) At start-up of the measuring instrument, a checksum over the legally relevant software is calculated and compared with a nominal value. The instrument only starts if the values match. Otherwise an event counter is increased by 1. During an update, the nominal value is modified to match the new software. The event counter value is retained and treated by the new software in the same manner as before.

- 6.3.8.4.5** Technical means shall be employed to guarantee the authenticity of the loaded software, i.e. that it originates from the owner of the certificate.

Example:

(II) The authenticity check is accomplished by cryptographic means such as a public key system. The owner of the certificate (in general the manufacturer of the measuring instrument) generates an electronic signature of the revised software or module using the *private key* in the manufactory. The *public key* is stored in a legally relevant software module of the measuring instrument receiving the signed revised software. The signature is checked using the *public key* when loading the revised software into the measuring instrument. If the signature of the loaded software is OK, it is installed and activated; if it fails the check, the loaded revised software is discarded, and the instrument continues to operate with the current version of the software or switches to an inoperable mode.

- 6.3.8.4.6** Technical means shall be employed to ensure the integrity of the loaded software, i.e. that it has not been inadmissibly changed before loading. This can be accomplished by adding a checksum or hash code of the loaded software and verifying it during the loading procedure.

- 6.3.8.4.7** If the loaded software fails the integrity test (6.3.8.4.6) or the authenticity test (6.3.8.4.5), the instrument shall discard the new version and use the previous version of the software or switch to an inoperable mode. In this mode, the measuring functions shall be inhibited. It shall only be possible to resume the download procedure, or to show an error.

- 6.3.8.4.8** An audit trail shall be employed to ensure that traced updates of the legally relevant software are adequately traceable within the instrument for subsequent verification and surveillance or inspection.

The audit trail shall contain at minimum the following information:

- success/failure of the update procedure;
- software identification of the installed version;
- software identification of the previous installed version;
- time stamp of the event;
- identification of the uploading party, i.e. the source of the update, if available.

An entry is generated for each update attempt regardless of the success.

The storage device that supports the traced update shall have a sufficient capacity to ensure the traceability of traced updates of the legally relevant software between at least two successive verifications of a measuring instrument in the field/inspections. After having reached the limit of the storage for the audit trail, it shall be ensured by technical means that further downloads are impossible without breaking a seal. See also 6.2.3.6.

*Note:* This requirement enables inspection authorities, which are responsible for the metrological surveillance of legally controlled instruments, to back-trace traced updates of the legally relevant software over an adequate period of time (depending on national legislation).

- 6.3.8.4.9** If the audit trail has no more capacity (see 6.3.8.4.8), an appropriate response is required, i.e. either the oldest entry may be deleted or the update procedure should not start at all.

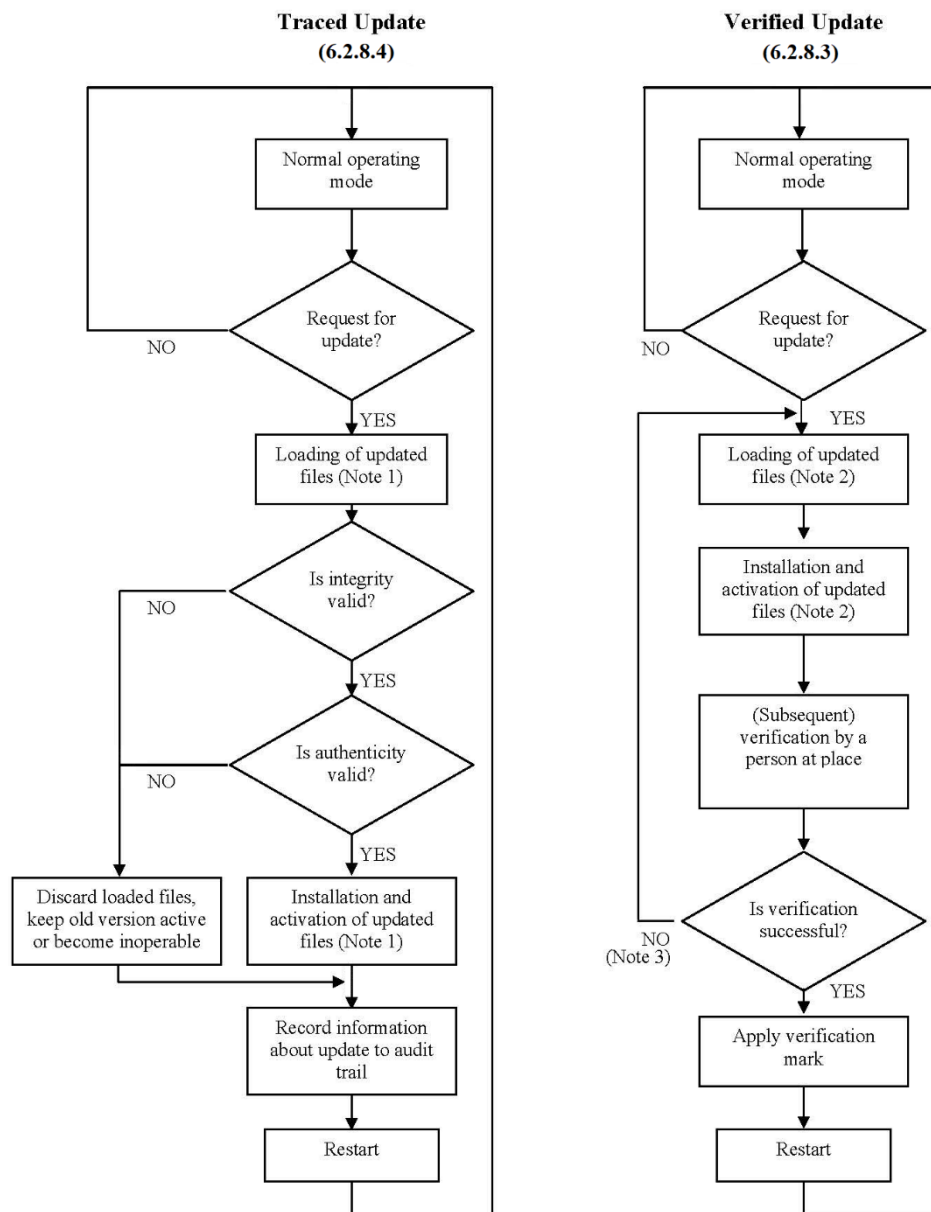


Figure 1 – Software update procedure

*Note 1:* In the case of a traced update, updating is separated into two steps: “loading” and “installing/activating”. This implies that the software is temporarily stored after loading without being activated because it shall be possible to discard the loaded software and revert to the old version, if the checks fail.

*Note 2:* In the case of a verified update, the software may also be loaded and temporarily stored before installation, but depending on the technical solution loading and installation may also be accomplished in one step.

*Note 3:* Here, only failure of the verification of a measuring instrument due to the software update is considered. Failure due to other reasons does not require re-loading and re-installing of the software, symbolised by the NO-branch.

**6.3.8.4.10** When the software is updated, the audit trail should not be erased or overwritten.

## **6.3.9 Remote Verification Capability**

### **6.3.9.1 General**

In case the instrument facilitates remote verification, the following requirements shall apply.

*Note:* There shall be a description of the remote verification procedure for accessing/reading of remote verification data and for executing remote verification procedures, and that shall be made available to the relevant authorities depending on national legislation, see clause 7.1.2.

**6.3.9.1.1** The modules involved in the remote verification procedure are part of the legally relevant software and shall fulfil the relevant requirements.

**6.3.9.1.2** It shall always be possible to establish and ensure the integrity of the instrument to be verified.

*Note:* This requirement specifically also applies to the legally relevant software which sends data, including the audit trail.

**6.3.9.1.3** It shall be possible to establish the authenticity of the instrument, i.e. the instrument shall be uniquely identified and other means shall be provided to ensure authenticity.

*Note:* This requirement specifically also applies to the legally relevant software which sends data, including the audit trail.

#### **Example:**

- (II) An instrument uses an asymmetric key pair to establish its authenticity prior to remote verification: The requesting (verification) party sends a random number to the instrument, which is then digitally signed by means of a private key. The signed response is then checked with the known public key of the instrument. Only if the signature matches the public key, is communication established.

**6.3.9.1.4** The instrument shall store logging data, audit trails, and make these available for remote verification purposes.

**6.3.9.1.5** The instrument shall use time stamps according to 6.2.7, provide evidence of an intervention (6.2.3), audit trails (6.2.3.6) and shall have a facility for detection of significant defects (6.2.6.1) for the purpose of remote verification.

**6.3.9.1.6** An ongoing measurement shall not be influenced by remote verification.

**6.3.9.1.7** The use of the verification procedure shall not influence the compliance with other requirements.

Access to the verification procedures, specific test items or commands shall be available but can be restricted if these influence compliance with other requirements, such as:

- Requirements on battery life,
- on resources or
- delays in the measurement process.

**6.3.9.1.8** The software integrity of the instrument shall not be influenced by the remote verification procedure.

**6.3.9.1.9** There shall be a legally relevant interface for data extraction for remote verification purposes.

**6.3.9.1.10** Interfaces for remote verification shall be protected, see clause 6.3.2.1.2.

**6.3.9.1.11** Access rights to the instrument for remote verification data shall be described in the documentation and made available to the relevant authorities depending on national legislation, see clause 7.1.2.

**6.3.9.1.12** Provisions shall be made to securely store the result of the remote verification in the measuring instrument. This data shall be protected and secured. Securing needs to ensure that only the remote verification software has write permissions.

At minimum, a unique ID (at least identifying the verification authority) and the date of the verification shall be stored.

*Note 1:* Depending on the instrument type or the application, additional data may be required which is to be specified in the relevant Recommendation.

*Note 2:* The recognition of a verification mark and the data it contains are subject to national requirements. If not in compliance with national regulations, the manufacturer shall disable the remote verification functionality.

Stored results of the verification in the instrument shall comply with clause 6.3.4.

## **6.3.9.2 Specific remote verification procedures**

For specific remote verification procedures (see 8.3) the instrument shall fulfil the following requirements 6.3.9.3 to 6.3.9.4.

### 6.3.9.3 **Direct extraction of test items**

**6.3.9.3.1** When checking software integrity, the integrity measure (checksum, hash) shall be calculated immediately before transmitting the integrity measure to the remote verification software.

**6.3.9.3.2** Test items shall be uniquely identified. The obtained test items shall be unambiguously linked to the measuring instrument to be verified.

**6.3.9.3.3** Relevant test items identified by the PG (see 8.3.2) shall be available depending on the specific requirement to be tested and the instrument type (e.g. approved type number, serial number, legally relevant settings and parameters, verification information and status, software version identification, software integrity, audit logs/trails, change logs, event logs etc.).

*Note:* See clause 8.3.4 for examples of test items for a specific remote verification procedure.

### 6.3.9.4 **Live integrity verification**

**6.3.9.4.1** A means shall be implemented to verify connection requirements.



## 7 Type evaluation

### 7.1 Software documentation to be supplied for type evaluation

#### 7.1.1 General

For type evaluation the manufacturer of the measuring instrument shall declare and document all functions, relevant data structures and software interfaces of the legally relevant software that are implemented in the instrument. All commands and their effects shall be described completely in the software documentation to be submitted for type evaluation.

Furthermore, the application for type evaluation shall be accompanied by a document or other evidence that supports the assumption that the design and characteristics of the software of the measuring instrument comply with the requirements of the relevant Recommendation, in which the general requirements of this Document have been incorporated.

*Note:* In cases of dynamic modules of legally relevant software (e.g. evolving ML-models), the manufacturer shall describe clear ways of verification and evaluation of said dynamic modules. With respect to metrological performance testing more generally, PGs may need to consider the impact of dynamic modules of legally relevant software on traditional methods and assumptions regarding the interpolation or extrapolation of measurement performance across the operational range of the measuring instrument under evaluation and test.

#### 7.1.2 Contents of the documentation

The documentation (for each measuring instrument or component) shall at least include:

- description of the legally relevant software and how the requirements are met:
  - list of legally relevant software modules;
  - description of the software interfaces of the legally relevant software and of the commands and data flows via this interface;
  - depending on the evaluation method chosen in the relevant Recommendation (see 7.3 and 7.4) the source code shall be made available to the type evaluation authority if raised risk level is required by the relevant Recommendation;
  - list of parameters to be protected and description of protection means;
- description of suitable system configuration and minimal required resources (see 6.3.6);
- description of the security means of the operating system (password, etc. if applicable);
- description of the protective means;
- overview of the system hardware, e.g. topology block diagram, type of computer(s), type of network, etc. Where a hardware component is deemed legally relevant or where it performs legally relevant functions, this should also be identified;
- description of the accuracy of the algorithms (e.g. filtering of A/D conversion results, price calculation, rounding algorithms, etc.);
- description of the user interface, menus and dialogues;
- software identification and instructions for obtaining it from an instrument in use;

- list of commands of each hardware interface of the measuring instrument or component;
- if an internal clock is synchronized with legal time, the method of synchronization and traceability to legal time;
- list of durability errors that are detected by the software and, if necessary for understanding, a description of the detecting algorithms;
- the required metadata for legally relevant measurement data;
- description of datasets stored or transmitted;
- if detection of significant defects is realised in the software, a list of significant defects that are detected and a description of the detecting algorithm;
- if fault detection is realised in the software, a list of faults that are detected and a description of the detecting algorithm;
- if an audit trail is realised in the software, a description of how to access the audit trail;
- if remote verification is supported
  - a description of the remote verification procedure for accessing/reading of remote verification data and for executing remote verification procedures with an explanation how a certain test item can be used to evaluate if a certain requirement is fulfilled;
  - description of the access rights to the instrument for remote verification and a description how test items can be obtained (and made available to relevant authorities depending on national legislation);
- if dynamic modules of legally relevant software are present
  - a description of the prioritization of using all legally relevant parts, including dynamic modules of legally relevant software;
  - a description of the means to validate the conformity of devices in use even in presence of dynamic parameter changes;
  - detailed description of the dynamic module's algorithm design as well as a description of the training process and the used training datasets
- the operating manual.

## 7.2 Requirements for the evaluation procedure

### 7.2.1 General

In the framework of type evaluation, test procedures are based on well-defined test setups and test conditions and can rely on metrologically traceable comparative measurements. The accuracy or correctness of software in general cannot be measured in a metrological sense, though there are standards that prescribe how to “measure” software quality [e.g. ISO/IEC 25040:2011 series [5]]. The procedures described here take into consideration both the legal metrology needs and also well-known evaluation and verification methods in software engineering, but which do not have the same goals (e.g. a software developer who searches for errors but who also optimises performance). As shown in 7.4, each software requirement needs individual adaptation of suitable evaluation procedures. The effort for the procedure should reflect the risk level.

The aim is to verify the fact that the instrument to be certified complies with the requirements of the relevant Recommendation. For software-controlled instruments the evaluation procedure comprises examinations, analysis, and tests and the relevant Recommendation shall include an appropriate selection of methods described below.

The methods described below focus on the type evaluation. Verifications of every single instrument in use in the field are not covered by those evaluation methods. Refer to clause 8 *Verification of a measuring instrument* for more information.

The methods specified for software evaluation are described in 7.3. Combinations of these methods forming a complete software evaluation procedure adapted to all requirements defined in clause 6 are specified in 7.4.

The manufacturer shall attest that no hidden or undocumented properties exist. (e.g. parameters, commands, functions, backdoors.)

This Document does not ask manufacturers for extra declarations that documentation is correct and complete. However, any country may require this declaration, as a part of the specified software examination process.

### 7.2.2

#### **Information to be included in the certificate**

The following information shall be included in the certificate:

- software identification of all certified versions;
- method to display the current software identification on the certified instrument in use;
- securing means as well as means to provide evidence of an intervention and the method to check them (e.g. hardware seals, event counters, audit trails.);
- software modules under legal control, including whether or not the instrument is equipped with a remote verification procedure;
- if applicable:
  - means of integrity protection checking;
  - software operating environment,
  - test items with their unique identification used for the remote verification procedure.

## 7.3 Verification and evaluation methods

### 7.3.1 Overview of methods and their application

The selection and sequence of the following methods are not prescribed and may vary in a software evaluation procedure from case to case.

This is a rough overview. For more details, see 7.3.2.

Table 1 – Overview of the proposed selected verification and evaluation methods

Abbreviation	Description	Application	Preconditions, tools for application	Special skills for performing
AD	Analysis of the documentation and evaluation of the design (7.3.2.1)	Always	Documentation	-
VFTM	Verification by functional testing of metrological functions (7.3.2.2)	Correctness of the algorithms, uncertainty, compensating and correcting algorithms, rules for price calculation	Documentation, specimen	-
VFTSw	Verification by functional testing of software functions (7.3.2.3)	Correct functioning of communication, indication, evidence of intervention, protection against operating errors, protection of parameters, detection of significant defects	Documentation, specimen	-
DFA	Metrological data flow analysis (7.3.2.4)	Software separation, evaluation of the impact of commands on the instrument's functions	Source code, tools for analysing source code	Knowledge of programming languages
CIWT	Code inspection and walkthrough (7.3.2.5)	All purposes	Source code, tools for analysing source code	Knowledge of programming languages
SMT	Software module testing (7.3.2.6)	All purposes when input and output can clearly be defined	Source code, testing environment	Knowledge of programming languages

Table 2 – Recommendations for combinations of evaluation and verification methods for the various software requirements (acronyms defined in Table 1)

Requirement		Examination level A (normal examination level)	Examination level B (extended examination level)	Comment
<b>General requirements</b>				
6.2.1	Software identification	AD + VFTSw	AD + VFTSw + CIWT	Select “B” if high conformity is required
6.2.2	Correctness of algorithms and functions	AD + VFTM	AD + VFTM + CIWT/SMT	
<b>Software protection</b>				
6.2.3	Evidence and prevention of intervention	AD + VFTSw	AD + VFTSw	
6.2.4	Prevention of misuse	AD + VFTSw	AD + VFTSw + DFA/CIWT/SMT	Select “B” in case of high risk of fraud
6.2.5	Demands on the user	AD + VFTSw	AD + VFTSw	
<b>Support of hardware features</b>				
6.2.6.1	Detection of significant defects	AD + VFTSw	AD + VFTSw + CIWT + SMT	Select “B” if high reliability is required
6.2.6.2	Durability protection	AD + VFTSw	AD + VFTSw + CIWT + SMT	Select “B” if high reliability is required
6.2.6.3	Information for remote verification	AD + VFTSw	AD + VFTSw + CIWT + SMT	Select “B” if high reliability is required
6.2.7	Time stamps	AD + VFTSw	AD + VFTSw + SMT	
<b>Specification and separation of legally relevant components and modules and requirements for interfaces</b>				
6.3.2.1	Separation of components	AD	AD + DFA/CIWT	
6.3.2.2	Separation of software modules	AD	AD + DFA/CIWT	
6.3.3	Shared indications	AD + VFTM/ VFTSw	AD + VFTM/ VFTSw + DFA/CIWT	
6.3.4	Storage of data	AD + VFTSw	AD + VFTSw + CIWT/SMT	Select “B” if storage of measurement data in unsecure storages is foreseen
6.3.4.2	Completeness of stored data	AD + VFTSw	AD + VFTSw + CIWT/SMT	Select “B” in case of high risk of fraud
6.3.4.3	Protection of stored data	AD + VFTSw	AD + VFTSw + SMT	
6.3.4.4	Automatic storing	AD + VFTSw	AD + VFTSw + SMT	

Requirement		Examination level A (normal examination level)	Examination level B (extended examination level)	Comment
6.3.5	Data transmission	AD + VFTSw	AD + VFTSw + CIWT/SMT	Select “B” if transmission of measurement data in open system is foreseen
6.3.5.2	Completeness of transmitted data	AD + VFTSw	AD + VFTSw + CIWT/SMT	Select “B” in case of high risk of fraud
6.3.5.3	Protection of transmitted data	AD + VFTSw	AD + VFTSw + SMT/	
6.3.5.4	Transmission delay or interruption	AD + VFTSw	AD + VFTSw + SMT	Select “B” in case of high risk of fraud, e.g. transmission in open systems
6.3.6	Compatibility of operating systems and hardware	AD + VFTSw	AD + VFTSw + SMT	
6.3.6.2	Hardware interfaces	AD + VFTSw	AD + VFTSw + SMT	
6.3.6.3	Boot process	AD + VFTSw	AD + VFTSw + SMT	
6.3.6.4	System resources	AD + VFTSw	AD + VFTSw + SMT	
6.3.6.5	Protection during use	AD + VFTSw	AD + VFTM/ VFTSw + DFA	
6.3.6.6	Communication with the legally relevant software	AD + VFTSw	AD + VFTM/ VFTSw + DFA	
6.3.6.7	Identification and traceability	AD + VFTSw	AD + VFTSw + SMT	
6.3.6.8	Suitable environment	AD + VFTSw	AD + VFTSw + SMT	
6.3.6.9	Constraints for operation	AD + VFTSw	AD + VFTSw + SMT	
<b>Maintenance and re-configuration</b>				
6.3.8.3	Verified update	AD	AD	
6.3.8.4	Traced update	AD + VFTSw	AD + VFTSw + CIWT/SMT	Select “B” in case of high risk of fraud
6.3.9.1	General	AD + VFTSw	AD + VFTSw + CIWT/SMT	
6.3.9.3	Direct extraction of test items	AD + VFTSw	AD + VFTSw + CIWT/SMT	
6.3.9.4	Live integrity verification	AD + VFTSw	AD + VFTSw + CIWT/SMT	

## 7.3.2 Description of selected verification and evaluation methods

### 7.3.2.1 Analysis of Documentation and Specification and Evaluation of the Design (AD)

Application:

Basic procedure for software evaluation.

Preconditions:

The procedure is based on the manufacturer's documentation of the measuring instrument. This documentation shall have a scope which is adequate for the application:

- 1) Specification of the externally accessible functions of the instrument in a general form (suitable for simple instruments with no interfaces except a display, all features verifiable by functional testing, low risk of fraud).
- 2) Specification of the software functions and interfaces (necessary for instruments with interfaces and for instrument functions that cannot be functionally tested and in case of increased risk of fraud). The description shall make evident and explain all software functions that may have an impact on the legally relevant features.

*Note:* In cases of dynamic modules of legally relevant software, documentation of the software functions shall include a detailed description of the dynamic module's algorithm design (e.g. the topology of the neural network and a description of its learning facility) as well as a description of the training process (e.g. training, validating, and testing) and the used training datasets, enabling assessment of the algorithm's compliance with the relevant Recommendation.

- 3) Concerning interfaces, the documentation shall include a complete list of commands or signals that the software is able to interpret. The effect of each command shall be documented in detail. The way shall be described in which the instrument reacts on commands that are not described in the documentation.
- 4) Additional documentation of the software for complex measuring algorithms, cryptographic functions, or crucial timing constraints shall be provided, if necessary for understanding and evaluating the software functions.

A general precondition for examination is the completeness of the documentation and the clear identification of the EUT, i.e. of the software packages that contribute to the legally relevant functions (see 7.1.2).

Description:

The examiner evaluates the functions and features of the measuring instrument using the documentation and decides whether they comply with the requirements of the relevant Recommendation. Metrological requirements as well as software-functional requirements defined in clause 6 (e.g. evidence of intervention, protection of adjustment parameters, disallowed functions, communication with other devices, update of software, detection of significant defects, etc.) shall be considered and evaluated. This task may be supported by the Software Evaluation Report Format (see Annex B).

Result:

The procedure gives a result for all characteristics of the measuring instrument, provided that the appropriate documentation has been submitted by the manufacturer. The result should be documented in a clause related to software

in a Software Evaluation Report (see Annex B) included in the Evaluation Report Format of the relevant Recommendation.

Complementary procedures:

Additional procedures should be applied, if examining the documentation cannot provide substantiated evaluation results. In most cases “Verifying the metrological functions by functional testing” (see 7.3.2.2) is a complementary procedure.

Reference:

IEC 61508-5:2010 [7].

### 7.3.2.2 **Verification by Functional Testing of the Metrological Functions (VFTM)**

Application:

For verifying correctness of algorithms for calculating the measurement result from measurement data, for linearisation of a characteristic, compensation of environmental influences, rounding in price calculation, etc.

Preconditions:

Operating manual, functioning specimen, metrological references, test equipment, test cases, instructions for test equipment.

When it is not clear how to verify a function of a software module, the onus to develop a test method should be placed on the manufacturer. In addition, the services of the programmer should be made available to the examiner for the purposes of answering questions.

Description:

Most of the evaluation and verification methods described in Recommendations are based on reference measurements under various conditions. Their application is not restricted to a certain technology of the instrument. Although it does not aim primarily at verifying the software, the test result can be interpreted as a verification of some software modules, in general even the metrologically most important. If the tests described in the relevant Recommendation cover all the metrologically relevant features of the instrument, the corresponding software can be regarded as being verified. In general, no additional software analysis or test needs to be applied to verify the metrological features of the measuring instrument.

*Note:* In cases of dynamic modules of legally relevant software, functional tests can only be performed on a snapshot of the dynamic legally relevant software modules. Even for such snapshots the examiner should check the outcome of the dynamic module’s algorithm under different circumstances to ensure the outcome of parameter corrections.

Result:

Algorithms are correct or not correct. Measurement results under all conditions are within the maximum permissible error (MPE) or not.

Complementary procedures:

The method is normally an enhancement of 7.3.2.1. In certain cases, it may be easier or more effective to combine the method with examinations based on the source code (7.3.2.5) or by simulating input signals (7.3.2.6) e.g. for dynamic measurements.



**References:**

Various specific Recommendations.

**7.3.2.3 Verification by Functional Testing of the Software Functions (VFTSw)****Application:**

For evaluation of e.g. protection of parameters, indication of a software identification, software supported detection of significant defects, configuration of the system (especially of the software environment), etc.

**Preconditions:**

Operating manual, software documentation, functioning specimen, test equipment, test cases, instructions for test equipment.

When it is not clear how to verify a function of a software module, the onus to develop a test method should be placed on the manufacturer. In addition, the services of the programmer should be made available to the examiner for the purposes of answering questions.

**Description:**

Required features described in the operating manual, instrument documentation or software documentation are checked practically. If they are software-controlled, they are to be regarded as verified if they function correctly without any further software analysis. Features addressed here are e.g.:

- normal operation of the instrument, if its operation is software-controlled. All switches or keys and described combinations should be employed and the reaction of the instrument evaluated. In graphical user interfaces, all menus and other graphical elements should be activated and checked;
- effectiveness of parameter protection may be checked by activating the protection means and trying to change a parameter;
- effectiveness of the protection of stored data may be checked by changing some data in the file and then checking whether this is detected by the software;
- indication of the software identification may be verified by practical checking;
- if detection of significant defects is software supported, the relevant software modules may be verified by provoking, implementing or simulating a fault and checking the correct reaction of the instrument;
- protection means that there is evidence of an intervention if changes are made to software, parameters, audit trails, etc. This can be tested by making changes and checking if this leads to evidence of an intervention.

**Result:**

Software-controlled feature under consideration is acceptable or not acceptable.

**Complementary procedures:**

Some features or functions of a software-controlled instrument cannot be practically verified as described. If the instrument has interfaces, it is in general not possible to detect unauthorised commands only by trying commands at random. Besides that, a sender is needed to generate these commands. For the normal examination level method in 7.3.2.1 may cover this requirement.

For the extended examination level, a software analysis such as 7.3.2.4 or 7.3.2.5 is necessary.

References:

WELMEC Guide 2.3, Section 3 [8]; WELMEC Guide 7.2, Sections 4.2 and 5.2[9].

#### 7.3.2.4 **Metrological Dataflow Analysis (DFA)**

Application:

For analysis of the software design concerning the control of the data flow of measurement information through the data domains that are subject to legal control, including the examination of the software separation.

Preconditions:

Software documentation, source code, editor, text search program or special tools. Knowledge of programming languages.

Description:

It is the aim of this method to find all software modules that are involved in the calculation of the measurement result or that may have an impact on it. Starting from the hardware port where raw data from the sensor are available, the subroutine that reads them is searched. This subroutine will store them in a variable after possibly having done some processing. From this variable the intermediate value is read by another subroutine and so forth until the completed measurement result is output to the display. All variables that are used as storage for intermediate measurement data and all subroutines processing and transporting these data can be found in the source code simply by using a text editor and a text search program to find all other occurrences of the variable or the subroutine name.

Other data flows can be found by this method, e.g. from software interfaces to the interpreter of received commands. Furthermore, circumvention of a software interface (see 6.3.2.2) can be detected.

Result:

It can be verified whether software separation according to 6.3.2.2 is acceptable or not acceptable.

It can be verified whether the documented list of commands for each interface is complete or not.

Complementary procedures:

This method is recommended if software separation is realised and if high conformity or strong protection against manipulation is required. It is an enhancement to 7.3.2.1-7.3.2.3 and to 7.3.2.5.

Reference:

IEC 61131-3.

#### 7.3.2.5 **Code Inspection and Walk Through (CIWT)**

Application:

Any feature of the software may be verified with this method if extended examination intensity is necessary.

**Preconditions:**

Source code, text editor, tools. Knowledge of programming languages.

**Description:**

The examiner walks through the source code assignment by assignment, evaluating the respective part of the code to determine whether the requirements are fulfilled and whether the functions and features are in compliance with the documentation.

The examiner may also concentrate on algorithms or functions that he has identified as complex, error-prone, insufficiently documented, etc. and inspect the respective part of the source code by analysing and checking.

Prior to these examination steps the examiner will have identified the legally relevant software modules, e.g. by applying the metrological data flow analysis (see 7.3.2.4). In general code inspection or walk through is limited to this part.

*Note:* Any static analysis can only examine a snapshot of the dynamic modules of legally relevant software.

**Result:**

Implementation compatible with the software documentation and in compliance with the requirements or not.

**Complementary procedures:**

This is an enhanced method, additional to 7.3.2.1 and 7.3.2.4. Normally it is only applied in spot checks.

**Reference:**

IEC 61508-5:2010 [7].

### 7.3.2.6 **Software Module Testing (SMT)**

**Application:**

This method is only used in exceptional cases. It is applied when functions of a software module cannot be examined exclusively on the basis of written information. It is appropriate and effective in the verification of dynamic measurement algorithms.

**Preconditions:**

Source code, development tools, functioning environment of the software module under test, input dataset and corresponding nominal output dataset or tools for automation. Skills in information technology, knowledge of programming languages. Co-operation with the programmer of the module under test is advisable.

**Description:**

The software module under test is integrated in a test environment, i.e. a specific test program that calls the module under test and provides it with all necessary input data. The test program receives actual output data from the module under test and compares them with the nominal values.

**Result:**

Module under test is correct or not.

**Complementary procedures:**

This is an enhanced method, additional to 7.3.2.2 or 7.3.2.5.

Reference:

IEC 61508-5:2010 [7].

## **7.4 Software evaluation procedure**

The software evaluation procedure consists of a combination of evaluation and verification methods. The relevant Recommendation may specify details concerning the software evaluation procedure, including

- a) which of the evaluation and verification methods described in 7.3 shall be carried out for the requirement under consideration,
- b) how the evaluation of test results shall be performed,
- c) which results should be included in the software test report, which results should be included in the evaluation report and which results should be integrated in the certificate (see Annex B).

In Table 2 two alternative examination levels Normal (A) and Extended (B) for the software evaluation procedures are defined. DFA, CIWT and SMT methods are only suggested for level B. Level B implies an extended examination compared to A. The selection of level B shall be justified by the PGs together with evidence of mitigated risk. A selection between A and B examination levels may be made in the relevant Recommendation – different or equal for each requirement – in accordance with the expected

- risk of fraud,
- area of application,
- required conformity to certified type, and
- risk of wrong measurement result due to operating errors.

See clause 4 for preliminary guidance on risk assessment.

## **7.5 Equipment Under Test (EUT)**

Normally, tests are carried out on the complete measuring instrument (functional testing). If the size or configuration of the measuring instrument does not lend itself to testing as a whole unit or if only a separate component or software module of the measuring instrument is concerned, the relevant Recommendation may indicate that the tests, or certain tests, shall be carried out on the components or software modules separately, provided that, in the case of tests with the components or software modules in operation, these are included in a simulated setup, sufficiently representative of its normal operation. The applicant is responsible for the provision of all the required equipment and specimens.

## 8 Verification of a measuring instrument

### 8.1 General

If metrological control of measuring instruments is prescribed in a country, there shall be means to check in use during operation the identity of the software, the validity of parameter adjustments and the conformity to the certified type.

The relevant Recommendation may require carrying out the verification of the software in one or more stages according to the nature of the considered measuring instrument.

The verification of the software shall include

- an examination of the conformity of the software to verify that it is the certified version (e.g. check of the software identification, check of securing means and protection means),
- an examination of the configuration to verify that it is compatible with the declared minimal configuration, if given in the certificate,
- an examination of the inputs/outputs of the measuring instrument to verify that they are free of inadmissible influence, and
- an examination of the device-specific parameters (especially the adjustment parameters) to verify that they are correctly set and a check of the securing and protection means to check the integrity of the parameters.

PGs shall consider the following subclause when writing instrument-specific verification procedures. The methods given in 8.2 are proposed as the standard procedure.

*Note:* National authorities may seek to develop a set of distinct (proprietary) data set types for use in testing and validation once devices are deployed in the field. This could be particularly applicable to dynamic modules of legally relevant software.

### 8.2 Verification methods, test items

The following methods comprise the verification steps which are needed to check the requirements of 6.1 and 6.3. The aspects in 8.2.1 to 8.2.4 shall be examined by the instructions listed in the corresponding subclause below.

#### 8.2.1 Documents

The initial step of any software verification shall consist of checking the EUT for compliance with the certificate and its annexes:

- check whether the certificate is valid;
- check whether the EUT complies with the pattern as described in the certificate and its annexes;
- check whether the operating manual is available (if required).

#### 8.2.2 Integrity of the software

Software integrity may be checked in one of two ways:

- indirectly: Check whether all seals required in the certificate are set at the right place and are intact;
- directly: Check the software identifiers as required in the certificate.

*Note:* The second item overlaps with the first item of 8.2.4.

**Example:**

Calculation of a checksum of the program code that is compared with the nominal value.

**8.2.3 Parameters****8.2.3.1 Correctness**

The correctness of parameters may be checked as follows:

- indirect metrological verification of parameters: Perform a measurement and compare the results with a reference;
- check whether all settable parameters are within the allowed range.

**8.2.3.2 Integrity**

The integrity of parameters may be checked as follows:

- check whether the seals protecting the parameters are intact;
- check the audit trail for entries concerning parameters.

**8.2.4 Identity of the software**

The identity of the software may be checked as follows:

- check that the software identifier provided by the EUT is specified as valid for use in the certificate;
- check the entries of the audit trail for traced updates (see 6.3.8.4.7).

*Note:* The first item overlaps with the second item of 8.2.2.

**8.3 Remote Verification****8.3.1 Introduction and limitations**

Remote verification encompasses a set of procedures to support verification of an instrument in the field, potentially without a person on site. During remote verification (see Figure 2), a remote unit [5] issues commands through a secure connection [2] to the device to be verified [1] by means of its verification interface [3]. The device will trigger one or more verification algorithms [4] internally and send their output back to the remote unit where they are checked, displayed [7] and logged [6].

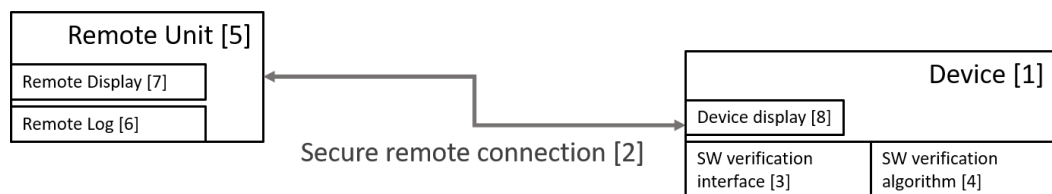


Figure 2 – Remote verification procedure

Remote verification procedures may be performed in one of two ways (depending on national legislation):

- 1) Completely: Check compliance of the measuring instrument with all the requirements remotely;

- 2) Partially: Check compliance of the measuring instrument covering only those requirements that can be evaluated remotely, in addition to checking compliance for the other requirements in situ.

Examples:

- 1) If it is possible to check compliance of the measuring instrument with all the requirements remotely (i.e. the measuring instrument is correctly installed; operating within MPEs; the integrity of that measuring instrument is intact, including the integrity of hardware seals; the readability requirements of the display are met: the display is not damaged); then no verification (or inspection) of the instrument needs to be carried out in situ (depending on national legislation).
- 2) If it is impossible to evaluate compliance with all the requirements remotely (i.e. only the evaluation of requirements such as the integrity of that measuring instrument can be performed remotely); then a partial verification (or inspection) of the instrument shall still be carried out in situ (depending on national legislation).

### 8.3.2 General requirements

Remote verification should cover the communication between legally relevant software modules, see 6.3.5. The communication connection between legally relevant software of the measuring instrument and software for verification purposes shall be available. This connection shall comply with clause 6.3.5, transmission via communication lines.

PGs shall define a list of relevant data for verification purposes depending on the instrument type (e.g. approved type number, , serial number, legally relevant settings and parameters, verification information and status, software version identification, software integrity, audit logs/trails, change logs, event logs etc.).

*Note 1:* The certificate shall state that remote verification is foreseen for this instrument.

*Note 2:* The requirements for remote verification will depend on national legislation.

*Note 3:* The device to be remotely verified needs to be available and ready.

*Note 4:* The device needs to be able to execute the verification procedures.

*Note 5:* D31 only imposes requirements on the measuring instrument's software, verification software is covered by national legislation.

The following clauses 8.3.3 to 8.3.6 describe examples of specific remote verification procedures and list the test items necessary for those remote verification procedures.

### 8.3.3 Extraction of audit trails or other logging mechanisms

#### 8.3.3.1 General

The purpose of this remote verification procedure is to check a measuring instrument's operational history. For that purpose, it is necessary to establish first the authenticity of the measuring instrument and the integrity.

After the authenticity and integrity has been established, retrieval of the relevant test items is initiated.

Applicable test items for this remote verification procedure are software integrity, identity of software, evidence of interventions, audit trails, detection of significant defects.

If applicable, the value of these test items is compared with a reference value.

- 8.3.3.2 A reference for all legally relevant software (measuring instrument software) shall be made available to the relevant authorities including approved type, serial number, legally relevant settings and parameters, verification information and status, software version identification, software integrity, audit logs/trails, change logs, event logs, etc. depending on national legislation

*Note:* Requirements on the external storage for legally relevant remote verification data for inspection authorities will depend on national legislation.

#### 8.3.4 **Direct extraction of test items**

##### 8.3.4.1 **General**

During remote verification, specific data objects are remotely retrieved from the measuring instrument. These data objects (such as a specific parameter or a software version number) are then compared with a known reference.

Applicable test items for this remote verification procedure are software integrity, correctness of parameters, identity of software

*Note 1:* A reference for all test item values (allowed range, specific value) needs to be available. This could either be a certificate or a protocol from a previous/initial verification.

*Note 2:* The manufacturer shall provide information about external SW for performing tests, see also 8.3.3.

- 8.3.4.2 Whenever this use case is applied, the audit trail of the legally relevant software shall be checked first to ensure that the correct software communicates with the external world, see 8.3.3.

#### 8.3.5 **Live integrity verification methods**

##### 8.3.5.1 **General**

The purpose of this remote verification procedure is to check the software integrity of the measuring instrument.

For that purpose, it is necessary to establish first the authenticity of the measuring instrument and the integrity.

After the authenticity and integrity has been established, retrieval of the relevant test item is initiated.

The applicable test item for this remote verification procedure is software integrity.

The value of the test item is compared with a reference value.



### 8.3.6 Test setup, simulation of measuring process

#### 8.3.6.1 General

The following subclauses 8.3.6.2 to 8.3.6.5 each detail a specific realization of this remote verification procedure for specific types of measuring instruments. These procedures shall be secured.

Access to diagnostics, build in weights etc. shall be restricted.

*Note 1:* The manufacturer shall describe the test procedure the result of which shall be made available to the relevant authorities depending on national legislation, see clause 6.3.9 and clause 7.1.2.

*Note 2:* The manufacturer shall describe the simulation procedure the result of which shall be made available to the relevant authorities depending on national legislation, see clause 6.3.9 and clause 7.1.2.

8.3.6.2 Initiate an internal weighing procedure using a build-in weight in weighing instruments to determine the accuracy of the weighing algorithms in the weighing instrument.

The applicable test item for this remote verification procedure is the accuracy of weighing algorithms.

8.3.6.3 Initiate procedure using a build-in diagnostics facility to establish whether the current performance of a flow meter has degraded since last calibration and therefore whether a recalibration is needed.

Applicable test items for this remote verification procedure are the state of the instrument regarding durability, changes in fouling or aging.

8.3.6.4 Simulating a digital sensor and sending intermediate measuring results to the Digital Data Processing Unit and retrieve the measurement result to evaluate the accuracy of the measurement algorithms in the Digital Data Processing Unit.

The applicable test item for this remote verification procedure is the accuracy of the measurement algorithm in the Digital Data Processing Unit.

8.3.6.5 Simulating a starting signal to sensor at the beginning of a corridor of known length and sending starting time to the P2P mother unit. After a period, another simulation of an ending signal to sensor at the end of a corridor and retrieve the measurement result to evaluate the accuracy of the measurement algorithms in the mother unit.

The applicable test item for this remote verification procedure is the accuracy of the measurement algorithm in the mother unit.

## Annex A

### Bibliography (Informative)

At the time of publication, the editions indicated were valid. All referred documents are subject to revision, and the users of this Document are encouraged to investigate the possibility of applying the most recent editions of the referred documents indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

The actual status of the Standards referred to can also be found on the Internet:

IEC Publications: [http://www.iec.ch/searchpub/cur\\_fut.htm](http://www.iec.ch/searchpub/cur_fut.htm)

ISO Publications: <http://www.iso.org>

OIML Publications: <https://www.oiml.org/en/publications/>

(with free download of PDF files).

In order to avoid any misunderstanding, it is highly recommended that all references to Standards in International Recommendations and International Documents be followed by the version referred to (generally the year or date).

Ref.	Standards and reference documents	Description
[1]	OIML V 2-200:2012 International Vocabulary of Metrology – Basic and General Concepts and Associated Terms (VIM), 3rd Edition	Vocabulary, developed by the Joint Committee for Guides in Metrology (JCGM).
[2]	OIML D 11:2013 General requirements for measuring instruments – Environmental conditions	Guidance for establishing appropriate metrological performance testing requirements for influence quantities that may affect the measuring instruments covered by OIML Recommendations (EMC, climatic, mechanical influences).
[3]	ISO/IEC 9594-8:2017 Information technology -- Open Systems Interconnection -- The Directory: Part 8: Public-key and attribute certificate frameworks	ISO/IEC 9594-8:2017 specifies frameworks and a number of data objects that can be used to authenticate and secure the communication between two entities, e.g. between two directory service entities or between a web browser and a web server. The data objects can also be used to prove the source and integrity of data structures such as digitally signed documents.
[4]	ISO/IEC 2382-9:2015 Information technology -- Vocabulary -- Part 9: Data communication	Intended to facilitate international communication in data communication. Presents terms and definitions of selected concepts relevant to the field of data communication and identifies relationships among the entries.
[5]	ISO/IEC 25040:2011 series Information technology -- Software product evaluation	The ISO/IEC 25040:2011 series of Standards gives methods for measurement, assessment and evaluation of software product quality. They describe neither methods for evaluating software production processes nor methods for cost prediction (software product quality measurements may, of course, be used for both these purposes).

Ref.	Standards and reference documents	Description
[6]	OIML V 1:2013 International vocabulary of terms in legal metrology (VIML)	The VIML includes only the concepts used in the field of legal metrology. These concepts concern the activities of the legal metrology service, the relevant documents, as well as other problems linked with this activity. Also included in this Vocabulary are certain concepts of a general character which have been drawn from the VIM.
[7]	IEC 61508-5:2010 Functional safety of electrical/ electronic/ programmable electronic safety-related systems – Part 5: Examples of methods for the determination of safety integrity levels	Provides information on the underlying concepts of risk and the relationship of risk to safety integrity (see Annex A); a number of methods that will enable the safety integrity levels for the E/E/PE safety-related systems, other technology safety-related systems and external risk reduction facilities to be determined (see Annexes, B, C, D and E). Intended for use by Technical Committees in the preparation of Standards in accordance with the principles contained in IEC Guide 104 and ISO/IEC Guide 51.
[8]	WELMEC Guide 2.3, May 2005 Issue 3 Guide for Examining Software (Weighing Instruments)	This guideline specifies basic requirements to be applied to software for free programmable, PC-based modules or peripheral devices which are linked to, or form part of, NAWIs subject to legal control.
[9]	WELMEC Guide 7.2, Issue 2019 Software Guide (Measuring Instruments Directive 2014/32/EU)	This document provides guidance to all those concerned with the application of the Measuring Instruments Directive (European Directive 2014/32/EU; MID), especially for software-equipped measuring instruments. It addresses both manufacturers of measuring instruments and notified bodies which are responsible for conformity assessment of MID instruments. By following the Guide, compliance with the software- related requirements contained in the MID can be assumed.
[10]	ISO/IEC 27005:2018 Information technology -- Security techniques - - Information security risk management	This document provides guidelines for information security risk management. This document supports the general concepts specified in ISO/IEC 27001 and is designed to assist the satisfactory implementation of information security based on a risk management approach. Knowledge of the concepts, models, processes and terminologies described in ISO/IEC 27001 and ISO/IEC 27002 is important for a complete understanding of this document. This document is applicable to all types of organisations (e.g. commercial enterprises, government agencies, non-profit organisations) which intend to manage risks that can compromise the organisation's information security.

Ref.	Standards and reference documents	Description
[11]	IEEE 802.3-2018	Ethernet local area network operation is specified for selected speeds of operation from 1 Mb/s to 400 Gb/s using a common media access control (MAC) specification and management information base (MIB). The Carrier Sense Multiple Access with Collision Detection (CSMA/CD) MAC protocol specifies shared medium (half duplex) operation, as well as full duplex operation. Speed specific Media Independent Interfaces (MIIs) allow use of selected Physical Layer devices (PHY) for operation over coaxial, twisted pair or fiber optic cables, or electrical backplanes. System considerations for multisegment shared access networks describe the use of Repeaters that are defined for operational speeds up to 1000 Mb/s. Local Area Network (LAN) operation is supported at all speeds.
[12]	OIML D34:2019 Conformity to Type (CTT) – Pre-market conformity assessment of measuring instruments	This Document provides considerations for countries and economies, or Regional Legal Metrology Organisations (RLMOs), that are planning to develop conformity to type (CTT) programs in the field of legal metrology. This Document also provides illustrative examples of CTT programs currently in operation.
[13]	ISO 8601:2019	<p>The purpose of this document is to provide a standard set of date and time format representations for information interchange, in order to minimize the risk of misinterpretation, confusion and their consequences.</p> <p>This document specifies a set of date and time format representations utilizing numbers, alphabets and symbols defined in ISO/IEC 646. These representations are meant to be both human recognizable and machine readable.</p> <p>This document retains the most commonly used expressions for date and time of day and their representations from earlier International Standards in the field, including earlier editions of ISO 8601 and its predecessors.</p>

## Annex B

### Example of a software test report (Informative)

*Note:* The Technical Committees and Subcommittees developing OIML Recommendations should decide which information shall be included in Software Test Report, Evaluation Report and OIML Certificate of Conformity. E.g. the name, version and checksum of the executable code from the following example should be included in the Certificate.

#### Software Test report no XYZ122344

#### Evaluation of Software of the flow meter Tournesol Metering model TT100

The software of the measuring instrument was verified to show conformance with the requirements of OIML Recommendation R xyz.

The evaluation was based on OIML International Document D 31:2019, where the essential requirements for software are interpreted and explained. This report describes the evaluation of software needed to state conformance with the R-xyz.

Manufacturer	Applicant
Tournesol Metering	New Company
P.O. Box 1120333	Nova Street 123
100 Klow	1000 Las Dopicos
Syldavie	San Theodorod
Reference: Mr. Tryphon Tournesol	Reference: Archibald Haddock

#### Test object

The Tournesol Metering meter TT100 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.

The flow meter is described as a built-for-purpose device (an embedded system) with a storage device containing legally relevant data.

The flow meter TT100 is an independent instrument with a transducer connected. The transducer incorporates a temperature compensation. Adjustment of flow rates is possible by calibration parameters stored in a non-volatile memory of the transducer. It 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

**Tournesol Metering, P.O. Box 1120333, 100 Klow, Syldavie.**

The file name of the executable code is “**tt100\_12.exe**”.

The verified version of this software is **V1.2c**. The software version is presented on the display upon instrument start-up and by pressing the “level” button for 4 seconds.

The source code comprises the following legally relevant 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 executable code “**tt100\_12.exe**” is protected against modification by a checksum. The value of the checksum by algorithm XYZ is **1A2B3C**.

The evaluation was supported by the following documents from the manufacturer:

- TT 100 User Manual Release 1.6;
- TT 100 Maintenance Manual Release 1.1;
- Software description TT100 (internal design document, dated 22 Nov 2003);
- Electronic circuit diagram TT100 (drawing no 222-31, dated 15 Oct 2003).

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

### **Results of evaluation**

The evaluation was performed according to the OIML D 31:YYYY. The evaluation was performed between 1 November and 23 December 2003. A design review was held on 3 December by Dr. K. Fehler at Tournesol Metering head office in Klow. Other evaluation work was carried out at the National Testing & Measurement Laboratory by Dr. K. Fehler and Mr. S. Problème.

### **The following requirements were verified:**

- software identification;
- correctness of algorithms and functions;
- software protection;
- prevention against accidental misuse;
- evidence of intervention;
- support of hardware features;
- storage of data, transmission via communication systems.

### **The following evaluation and verification methods were applied:**

- analysis of the documentation and evaluation of the design;
- verification by functional testing of metrological features;
- walkthrough, code inspection;
- software module testing of module calculate.c with SDK XXX.

**Result**

The following requirements of the OIML D 31:YYYY were verified without any non-conformities being found:

6.2.1, 6.2.2, 6.2.3, 6.2.3.6, 6.3.4 and 6.3.5.

The result applies to the tested item with Serial No. 1188093-B-2004 only.

**Conclusion**

The software of the **Tournesol Metering TT100 V1.2c** fulfils the requirements of OIML R xyz.

National Testing & Measurement Lab.

Software Department

Signature(s):

Dr. K.E.I.N. Fehler

Technical manager

Mr. S.A.N.S. Problème

Technical Officer

Clause	Requirement	Passed	Failed	Remarks
<b>6.2</b> <b>6.2.1</b>	<b>General requirements</b> <b>Software identification</b> Software modules of a measuring instrument or component shall be unambiguously identified.			
<b>6.2.2</b>	<b>Correctness of algorithms and functions</b> The measuring algorithms and functions of a measuring instrument shall be appropriate and functionally correct for the given application and device type. The measurement result shall be displayed or printed correctly.			
<b>6.2.3</b> <b>6.2.3.1</b>	<b>Evidence and prevention of intervention</b> Software shall be protected against any changes, for example due to physical effects and intentional misuse, i.e. modification, loading or changes by swapping the memory device, unauthorised updates. During processing, measurement data shall be protected and secured. Software shall be protected in such a way that evidence of any intervention (e.g. software updates, parameters changes) shall be available. Mechanical sealing or software seals shall be used to protect measuring instruments or components. In case of dynamic modules of legally relevant software with predefined parameters, these shall be considered as a part of the software and treated as such. If necessary for the purpose of verification, data containing evidence of an intervention shall be displayed or printed on command and, if applicable, transmitted to the verification software.			
<b>6.2.3.2</b>	All inputs from the user interface shall be handled by a protective interface. Any function that can be activated by the user interface shall be clearly documented and not be able to influence the legally relevant characteristics of the instrument.			
<b>6.2.3.3</b>	All inputs from communication interfaces shall be handled by a protective interface. Any function that can be activated through a communication interface shall be clearly documented not be able to influence the legally relevant characteristics of the instrument remotely such as through a remote verification procedure or a software download .			
<b>6.2.3.4</b>	Legally relevant parameters shall be secured and protected in such a way that evidence of an intervention shall be available. If necessary for the purpose of verification of a measuring instrument, displaying or printing and, if applicable, transmitting the current relevant parameter settings to the verification software shall be possible. If the setting of certain device-specific parameters is available to the user the measuring instrument shall be fitted with a facility to automatically and non-erasably record any adjustment of the legally-relevant device-specific parameter.			
<b>6.2.3.5</b>	Software protection means shall comprise appropriate sealing by software and/or cryptographic means, making an intervention impossible or evident.			
<b>6.2.3.6</b>	Audit trails and event counters are part of the legally relevant software and shall be secured and protected as such. It shall not be possible to delete or inadmissibly change the data of the event counter or audit trails and it shall not be possible to exchange the audit trails or the value of the event counter when the software is updated. The audit trail or value of the event counter shall be displayed or printed on command and, if applicable, transmitted to the verification software.			
<b>6.2.4</b>	<b>Prevention of misuse</b> A measuring instrument shall be constructed in such a way that possibilities for unintentional, accidental, or intentional misuse are minimal.			
<b>6.2.5</b>	<b>Demands on the user</b> The software of a measuring instrument shall be designed in such a way that no unreasonable demands are required from the user to obtain a correct measurement result.			



Clause	Requirement	Passed	Failed	Remarks
<b>6.2.6</b> <b>6.2.6.1</b>	<b>Support of hardware features</b> <b>Detection of significant defects</b> The manufacturer of the instrument shall be required to design checking facilities into the software or hardware parts or provide means by which the hardware parts can be supported by the software parts of the instrument. If software is involved in the detection of significant defects, it shall appropriately act upon any detected defect.			
<b>6.2.6.2</b>	<b>Durability protection</b> If software is involved in durability protection, it shall appropriately act upon any detected durability error.			
<b>6.2.6.3</b>	<b>Information for remote verification</b> If support of 6.2.6.1 or 6.2.6.2 is part of the remote verification procedure it shall be possible to transmit data containing information in this respect to the verification software.			
<b>6.2.7</b>	<b>Time stamps</b> The time stamp shall be in a consistent format, allowing for easy comparison of two different records and tracking progress over time. If a measuring instrument uses time stamps, the instrument shall contain an internal clock which shall be used for the creation of the time stamp. Depending on the kind of instrument or on the field of application, setting the clock may be legally relevant and appropriate protection means shall be taken according to the risk level to be applied. Automatic setting of the time shall only be possible if legal time is used as a time base in an authenticated manner.			
<b>6.3</b> <b>6.3.2</b>	<b>Requirements specific for configurations</b> <b>Specification and separation of legally relevant parts and specification of interfaces</b> Legally relevant software modules or hardware components of a measuring instrument shall not be inadmissibly influenced by another device or by other modules or components of the measuring instrument.			

Clause	Requirement	Passed	Failed	Remarks
<b>6.3.2.1</b>	<b>Separation of components</b>			
<b>6.3.2.1.1</b>	Components of a measuring instrument that perform legally relevant functions shall be identified, clearly defined, and documented.			
<b>6.3.2.1.2</b>	A software-controlled component shall communicate with other components or devices through a protective interface. It shall not be possible to inadmissibly influence the legally relevant software, parameters or measurement data through these interfaces. It shall be demonstrated that the functions and data of legally relevant components cannot be inadmissibly influenced by commands received via the protective interface.			
<b>6.3.2.1.3</b>	If software seals are used to prevent components from being exchanged and pairing parameters are part of the seal, then these pairing parameters are legally relevant and shall be secured and protected in such a way that evidence of an intervention is available.			
<b>6.3.2.1.4</b>	PGs may decide that legally relevant components shall check the authenticity, integrity and/or availability of another software-controlled component. In case the authenticity and/or integrity check fails, or the other component is not available, the checking component shall appropriately act upon this.			
<b>6.3.2.1.5</b>	If a component is shared by multiple components, e.g. one display for multiple sensors, then all the components that share another component shall be unambiguously identified.			
<b>6.3.2.1.6</b>	In case the completeness of the measuring instrument cannot be visually checked (e.g. wireless or network-connected components), non-legally relevant software modules shall be prevented from calculation/presenting/spoofing the measurement result.			
<b>6.3.2.1.7</b>	In case legally relevant components with limited functionality and limited securing/protection capabilities are applied, they shall have limited access to the measurement data			
<b>6.3.2.2</b>	<b>Separation of software modules</b>			
<b>6.3.2.2.1</b>	The conformity requirement applies to the legally relevant software of a measuring instrument and it shall be made identifiable as described in 6.2.1.			
<b>6.3.2.2.2</b>	All legally relevant software modules shall communicate with other modules through a protective interface. It shall be demonstrated that the functions and data of modules that are legally relevant cannot be inadmissibly influenced by commands received via the protective interface. Measurement data shall not be made available to legally non-relevant modules prior to primary indication.			
<b>6.3.2.2.3</b>	There shall be an unambiguous assignment of each command to all initiated functions or data changes in the legally relevant software. Functions that are triggered through the protective software interface shall be declared and documented. Only documented functions shall be activated through the protective software interface.			
<b>6.3.2.2.4</b>	Where the legally relevant software has been separated from the non-relevant software part, the legally relevant software shall have priority using the resources over non-relevant software. The legally relevant process shall not be inadmissibly interrupted by legally non-relevant software. The measurement process shall not be delayed or blocked by other processes.			
<b>6.3.2.2.5</b>	When dynamic modules of legally relevant software have facilities for continuous learning that allow dynamic parameter changes during use, the manufacturer shall clarify the facilities and its priorities to the whole legally relevant software, especially in reference to the measuring functions. Where relevant, PGs may specify the requirement that the measuring functions shall not be inhibited/affected by the continuous learning process.			

Clause	Requirement	Passed	Failed	Remarks
<b>6.3.3</b>	<b>Shared indications</b> If a display or printout is used both for legally relevant and legally non-relevant outputs, the legally relevant information shall always be readable, and clearly distinguishable from other information.			
<b>6.3.4</b> <b>6.3.4.2</b>	<b>Storage of data</b> <b>Completeness of stored data</b> The stored measurement data shall include all relevant data necessary for future legally relevant use. Where measurement data is produced as a result of algorithms of dynamic modules of legally relevant software, the measurement data shall be marked or indicated as such. Such markings or indications and associated data shall form part of the legally relevant measurement data.			
<b>6.3.4.3</b>	<b>Protection of stored data</b> The stored measurement data shall be protected by appropriate means to guarantee authenticity, integrity. The software that displays or further processes the measurement data shall check the authenticity and integrity of the data after having read them from the storage. If an irregularity is detected an appropriate response shall be required. The storage component shall have sufficient permanency to ensure that the stored measurement data are not corrupted under normal storage conditions. Software modules that prepare data for storing, or that check data after reading are considered part of the legally relevant software. If appropriate, means shall be provided whereby cryptographic keys can only be input or read if a seal is broken.			
<b>6.3.4.4</b> <b>6.3.4.4.1</b>	<b>Automatic storing</b> When, considering the application, data storage is required, measurement data shall be stored automatically. A checking facility shall regularly check the availability of the storage and in the case the storage device is not available no measurement shall be possible. There shall be sufficient memory storage for the intended application. If automatic storage is required, no measurement shall be possible if the storage device is not available. When the data necessary for the calculation of the measurement result are relevant for legal purposes, all measurement result relevant data included in the calculation shall be automatically stored with the final value.			
<b>6.3.4.4.2</b>	Measurement data stored in a component to construct the measurement result can be deleted if the next module or component state a proper completion of expected actions engaged. Stored measurement data may be deleted if either the transaction is settled or these data are printed by a printing device subject to legal control.			
<b>6.3.5</b> <b>6.3.5.2</b>	<b>Data transmission</b> <b>Completeness of transmitted data</b> The transmitted measurement data shall include all relevant data necessary for future legally relevant use. Where measurement data is produced as a result of algorithms of dynamic modules of legally relevant software, the measurement data shall be marked or indicated as such. Such markings or indications and associated data shall form part of the legally relevant measurement data.			

Clause	Requirement	Passed	Failed	Remarks
<b>6.3.5.3</b>	<b>Protection of transmitted data</b> The transmitted data shall be protected by software means to guarantee the authenticity, integrity. The software that displays or further processes the measurement data shall check the authenticity, and integrity of the data received from a transmission channel. If an irregularity is detected an appropriate reaction shall be required. Means shall be provided whereby cryptographic keys used by cryptographic methods can only be input or read if a seal is broken.			
<b>6.3.5.4</b>	<b>Transmission delay or interruption</b> The measurement shall not be inadmissibly influenced by a transmission delay or interruption. If network services become unavailable or very slow, no measurement data shall be lost.			
<b>6.3.6</b> <b>6.3.6.2</b>	<b>Compatibility of operating systems and hardware</b> Hardware interfaces not equipped with a protective interface shall not be able to inadmissibly influence the legally relevant software, parameters or measurement data.			
<b>6.3.6.3</b> <b>6.3.6.3.2</b> <b>6.3.6.3.3</b> <b>6.3.6.3.4</b> <b>6.3.6.3.5</b>	<b>Boot process</b> The boot process shall ensure integrity and authenticity of the legally relevant software. If a chain of trust is established over the individual steps of the boot process to ensure 6.3.6.3.2, the processing of the chain of trust may be interrupted, as long as its integrity is preserved. The boot configuration shall be secured and protected. Bootling via open interfaces shall be prohibited.			
<b>6.3.6.4</b>	<b>System resources</b> The combination of the legally relevant software and the operating system shall ensure that there are enough resources for the operation of the legally relevant application.			
<b>6.3.6.5</b> <b>6.3.6.5.1</b> <b>6.3.6.5.2</b> <b>6.3.6.5.3</b> <b>6.3.6.5.4</b>	<b>Protection during use</b> The operation of software that is not legally relevant shall not inadmissibly influence the legally relevant application. The combination of the legally relevant software part and the operating system shall ensure that the legally relevant indication is distinguishable from other information. The access control feature of the operating system shall be configured in such way that the intended use cannot be inadmissibly influenced. The administration tasks of the legally relevant software shall be protected.			
<b>6.3.6.6</b>	<b>Communication with the legally relevant software</b> Communication with the legally relevant software part shall be made via protective interfaces. It shall be demonstrated that the legally relevant software, parameters, and data of components that are legally relevant cannot be inadmissibly influenced by commands received via the protective interface.			

Clause	Requirement	Passed	Failed	Remarks
<b>6.3.6.7</b> <b>6.3.6.7.1</b>  <b>6.3.6.7.2</b>	<b>Identification and traceability</b> The configuration of the operating system shall be identifiable. The identifier shall be displayed on command or during operation and, if applicable, transmitted to the verification software by the measuring instrument. Legally relevant configuration settings of the operating system shall be protected, i.e. changes to the legally relevant configuration shall be traceable.			
<b>6.3.6.8</b>	<b>Suitable environment</b> The manufacturer shall identify the hardware and software environment that is suitable. Minimum resources and a suitable configuration management necessary to guarantee correct functioning of the legally relevant software shall be declared by the manufacturer.			
<b>6.3.6.9</b>	<b>Constraints for operation</b> Technical means shall be provided to prevent operation, if the minimum resources or a suitable configuration are not met.			
<b>6.3.8</b> <b>6.3.8.3</b>	<b>Maintenance and reconfiguration</b> <b>Verified Update</b> The software to be updated may be loaded locally, i.e. directly on the measuring instrument, or remotely via a network. A protection measure (i.e. physical or electronic seal that must be broken for the update to take effect) provides evidence of an intervention.			

Clause	Requirement	Passed	Failed	Remarks
<b>6.3.8.4</b>	<b>Traced update</b>			
<b>6.3.8.4.2</b>	Depending on the needs and national legislation it may be necessary for the user or owner of the measuring instrument to give their consent. If so, the measuring instrument shall have a feature for the user or owner to express their consent prior to an update, e.g. by means of a push button.			
<b>6.3.8.4.3</b>	After initiation of the update procedure, a traced update of software shall run automatically. If some of the securing or protection measures of the instrument are turned off to enable updating, they shall be turned on again immediately after update, independent of the result of the update process.			
<b>6.3.8.4.4</b>	During a Traced update, any existing audit trail information and event counter values shall be retained.			
<b>6.3.8.4.5</b>	Technical means shall be employed to guarantee the authenticity of the loaded software.			
<b>6.3.8.4.6</b>	Technical means shall be employed to ensure the integrity of the loaded software, i.e. that it has not been inadmissibly changed before loading.			
<b>6.3.8.4.7</b>	If the loaded software fails the integrity test or the authenticity test, the instrument shall discard the new version and use the previous version of the software or switch to an inoperable mode. In this mode, the measuring functions shall be inhibited. It shall only be possible to resume the download procedure, or to show an error.			
<b>6.3.8.4.8</b>	An audit trail shall be employed to ensure that traced updates of the legally relevant software part are adequately traceable within the instrument for subsequent verification and surveillance or inspection. An entry is generated for each update attempt regardless of the success. The storage device that supports the traced update shall have a sufficient capacity to ensure the traceability of traced updates of the legally relevant software between at least two successive verifications of a measuring instrument in the field/inspections. After having reached the limit of the storage for the audit trail, it shall be ensured by technical means that further downloads are impossible without breaking a seal.			
<b>6.3.8.4.9</b>	If the audit trail has no more capacity, or the owner denies consent, an appropriate response is required, i.e. either the oldest entry may be deleted or the update should not start at all.			
<b>6.3.8.5</b>	When the software is updated, the audit trail should not be erased or overwritten.			

Clause	Requirement	Passed	Failed	Remarks
<b>6.3.9</b>	<b>Remote Verification Capability</b>			
<b>6.3.9.1</b>	<b>General</b>			
<b>6.3.9.1.1</b>	The modules involved in the remote verification procedure are part of the legally relevant software and shall fulfil the relevant requirements.			
<b>6.3.9.1.2</b>	It shall always be possible to establish and ensure the integrity of the instrument to be verified.			
<b>6.3.9.1.3</b>	It shall be possible to establish the authenticity of the instrument, i.e. the instrument shall be uniquely identified and other means shall be provided to ensure authenticity.			
<b>6.3.9.1.4</b>	The instrument shall store logging data, audit trails, and make these available for remote verification purposes.			
<b>6.3.9.1.5</b>	The instrument shall use time stamps according to 6.2.7, provide evidence of an intervention (6.2.3), audit trails (6.2.3.6) and shall have a facility for detection of significant defects (6.2.6.1) for the purpose of remote verification.			
<b>6.3.9.1.6</b>	An ongoing measurement shall not be influenced by remote verification.			
<b>6.3.9.1.7</b>	The use of the verification procedure shall not influence the compliance with other requirements.			
<b>6.3.9.1.8</b>	The software integrity of the instrument shall not be influenced by the remote verification procedure.			
<b>6.3.9.1.9</b>	There shall be a legally relevant interface for data extraction for remote verification purposes.			
<b>6.3.9.1.10</b>	Interfaces for remote verification shall be protected.			
<b>6.3.9.1.12</b>	Provisions shall be made to securely store the result of the remote verification in the measuring instrument. This data shall be protected and secured. Securing needs to ensure that only the remote verification software has write permissions.			
<b>6.3.9.3</b>	<b>Direct extraction of test items</b>			
<b>6.3.9.3.1</b>	When checking software integrity, the integrity measure (checksum, hash) shall be calculated immediately before transmitting the integrity measure to the remote verification software.			
<b>6.3.9.3.2</b>	Test items shall be uniquely identified. The obtained test items shall be unambiguously linked to the measuring instrument to be verified.			
<b>6.3.9.3.3</b>	Relevant test items identified by the PG shall be available depending on the specific requirement to be tested and the instrument type.			
<b>6.3.9.4</b>	<b>Live integrity verification</b>			
<b>6.3.9.4.1</b>	A means shall be implemented to verify connection requirements.			

**Annex C**  
**Remarks on measurement terminology**  
**(Informative)**

*Note:* This informative Annex is intended to illustrate the terms and definitions related to the measurement process and their usage in this OIML Document.

The definition of *Measurement Result*, in this document, is a "set of quantity values being attributed to a measurand together with the Measurement Result Relevant Data", this is illustrated in Figure A.1 as the Measured Quantity Value and Measurement Result Relevant Data, both being part of the Measurement Result.

And together with the Measurement Process Data these form the Measurement Data.

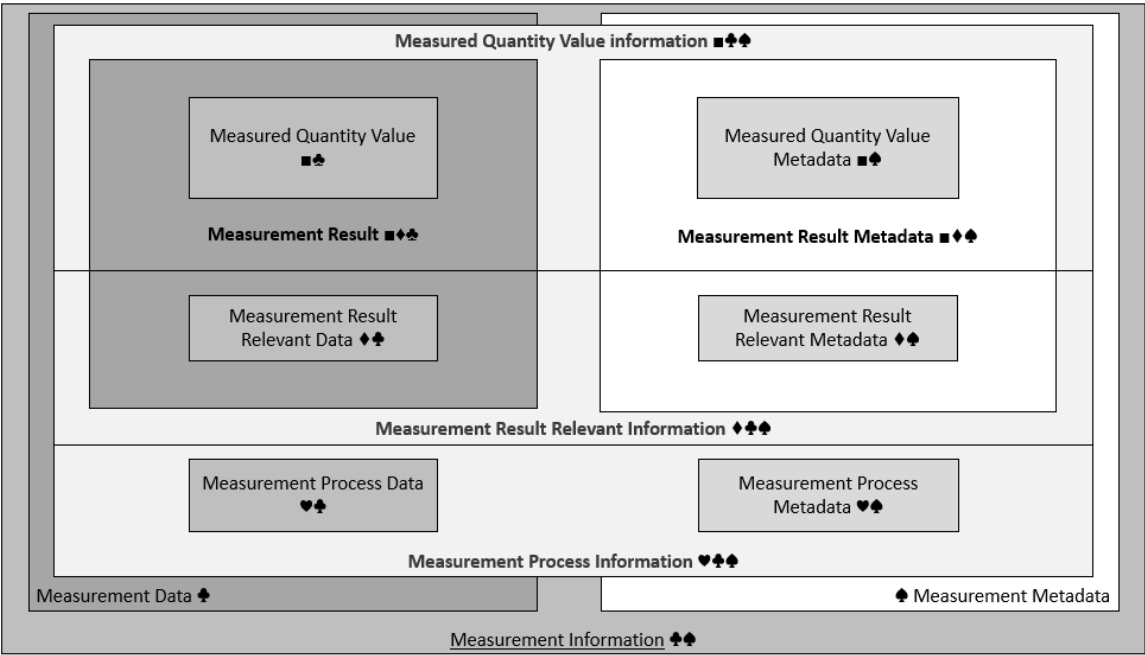


Figure A.1 – Visual representation of the Measurement Information

In general, this OIML Document distinguishes between measurement data and measurement metadata. If both are used together, measurement data are put into context; hence, measurement data plus measurement metadata equals measurement information.

This OIML Document also distinguishes between Measurement Result Relevant Information and Measurement Process Information.

Figure A.2 contains a flowchart to illustrate the distinction between the data relevant to the Measurement Result or data relevant to the Measurement Process.



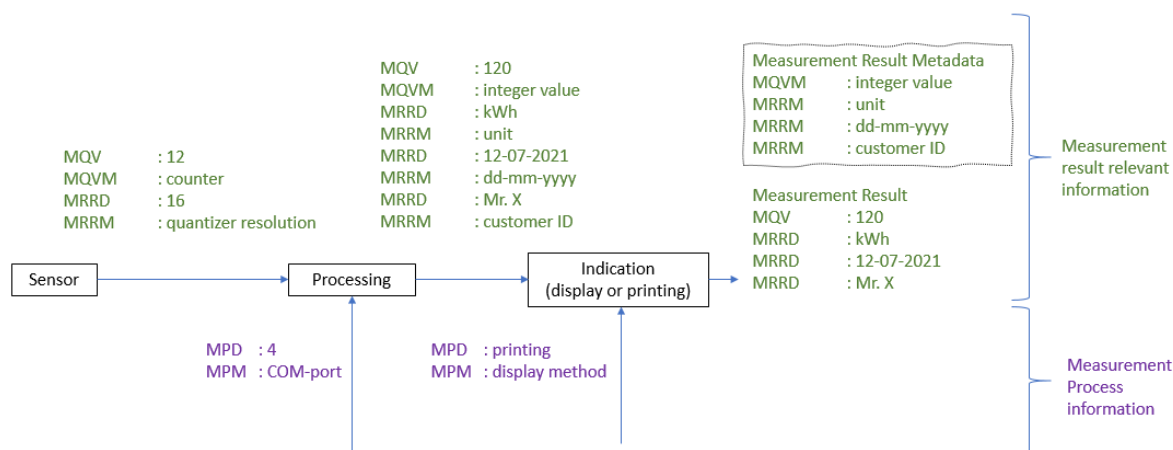


Figure A.2 – Flowchart of a measurement process, giving examples for the different data relevant to the Measurement Result or relevant to the Measurement Process.

Figure A.2. – Also indicates the data composing the Measurement Result: Measured Quantity Value (MQV) and the Measurement Result Relevant Data (MRRD), while the corresponding Measurement Result Metadata needed for the correct interpretation of the result is shown in a framed, dashed rectangle.

Figure A.2 shows a simple example of a measurement process. For each logical step (from data acquisition by the sensor to indication of the result) the following parts are noted:

- the Measured Quantity Value (MQV) and Measured Quantity Value Metadata (MQVM);
- the Measurement Result Relevant Data (MRRD) and the Measurement Result Relevant Metadata (MRRM);
- the Measurement Process Data (MPD) and the Measurement Process Metadata (MPM).

One strand of measurement information is related to the measurement result relevant information.

Data acquisition by the sensor delivers a raw counter value of 12 (MQV) with ‘counter’ as the Measured Quantity Value Metadata (MQVM) needed to interpret the data.

The Measurement Result Relevant Information (MRRD) is the ADC’s quantiser 16 bits resolution,

- where 16 is the Measurement Result Relevant Data (MRRD),
- while ‘quantiser resolution’ is the Measurement Result Relevant Metadata (MRRM), needed to interpret the data.

During processing, the Measured Quantity Value (MQV) with “integer value” as the Measured Quantity Metadata (MQMD) is assigned ‘kWh’ as Measurement Result Relevant Data (MRRD) with ‘unit’ as Measurement Result Relevant Metadata (MRRM), as well as a time stamp ‘17-07-2017’ (MRRD) with format ‘day-month-year’ (MRRM) and Mister X (MRRD) as customer ID (MRRM).

In both cases, during acquisition by the sensor and processing, the Measured Quantity Value (MQV) and Measurement Result Relevant Data (MRRD) form part of the Measurement Result, while the metadata is needed for the correct interpretation of the Measurement Result.

Another strand of measurement information is related to the measurement process: for acquisition of the Measured Quantity Value (MQV) from the sensor, COM-Port number 4 is used, where

- ‘4’ is the Measurement Process Data (MPD) and
- the ‘COM-Port’ is the Measurement Process Metadata (MPM) needed to understand the data element.

Indication of the result can be by means of a display or by printing.

The Measurement Process Data (MPD) ‘printing’ with the correspondent Measurement Process Metadata (MPM) ‘display method’ are both necessary for the measurement process, but they will not become part of the measurement result, nor the measurement result metadata.

It is up to the technical working groups to decide what Measurement Result Relevant Data is because under certain circumstances, Measurement Process Data (MPD) might become Measurement Result Relevant Data (MRRD).

In the given example, shown in Figure A.2, the COM-Port number 4 links the Measured Quantity Value (MQV) to a customer Mr. X, thus turning the Measurement Process Data (MPD) into Measurement Result Relevant Data (MRRD) during the processing step.

## Annex D

### Index

**Acceptable solution:** 6.2.1.

**Audit trail:** 3.2.1; 3.2.51; 6.2.3.1; 6.2.3.4; 6.2.3.5;  
6.2.3.6; 6.2.7; 6.3.2.1.3; 6.3.4.1; 6.3.6.5.4; 6.3.6.7.2;  
6.3.8.4.1; 6.3.8.4.4; 6.3.8.4.8; 6.3.8.4.9; 6.3.8.5;  
6.3.9.1.2; 6.3.9.1.3; 6.3.9.1.4; 6.3.9.1.5; 7.1.2; 7.2.2;  
7.3.2.3; 8.2.3.2; 8.2.4; 8.3.3; 8.3.3.1; 8.3.4.2.

**Authentication:** 3.2.2; 3.2.3; 6.3.8.4.1.

**Authenticity:** 3.2.3; 3.2.10; 3.2.16; 6.2.3.5; 6.3.2.1.4;  
6.3.4.3; 6.3.5.3; 6.3.6.3.2; 6.3.8.4.5; 6.3.8.4.7; 6.3.9.1.3;  
8.3.3.1; 8.3.5.1.

**Checking facility:** 3.2.5; 6.2.3.1; 6.2.3.4; 6.2.6.1;  
6.3.4.4.1.

**Command:** 3.2.56; 6.2.1; 6.2.3.1; 6.2.3.2; 6.2.3.6;  
6.3.2.1.2; 6.2.4; 6.3.2.1.2; 6.3.2.1.7; 6.3.2.2.2;  
6.3.2.2.3; 6.3.6.2; 6.3.6.6; 6.3.6.7.1; 6.3.9.1.7; 7.1.1;  
7.1.2; 7.2.1; 7.3.2.1; 7.3.2.3; 7.3.2.4; 8.3.1.

**Communication:** 3.2.7; 3.2.64; 5.2; 6.2.1; 6.2.3.3;  
6.3.2.1.3; 6.3.2.2.1; 6.3.2.2.26.3.2.2.3; 6.3.2.2.4;  
6.3.4.4.1; 6.3.6.6; 6.3.6.8; 6.3.9.1.3; 7.3.2.1; 8.3.2.

**Communication interface:** 3.2.7; 6.2.1; 6.2.3.3.

**Component:** 2.3; 3.2.7; 3.2.8; 3.2.12; 3.2.21; 3.2.28;  
3.2.29; 3.2.58; 3.2.66; 3.2.66; 3.2.68; 6.2; 6.2.1;  
6.2.3.1; 6.2.6.1; 6.2.6.2; 6.3.2; 6.3.2.1.1; 6.3.2.1.2;  
6.3.2.1.3; 6.3.2.1.4; 6.3.2.1.5; 6.3.2.1.6; 6.3.2.1.7;  
6.3.2.1.8; 6.3.2.2.1; 6.3.2.2.2; 6.3.3; 6.3.4.3;  
6.3.4.4.2; 6.3.5.4; 6.3.6.6; 6.3.7; 6.3.8.3; 6.3.8.4.1;  
7.1.2; 7.5.

**Cryptographic certificate:** 3.2.9; 3.2.16; 6.2.3.5;  
6.3.2.1.3.

**Cryptographic means:** 3.2.10; 6.2.3.5; 6.3.8.4.5.

**Data domain:** 3.2.11; 3.2.56; 3.2.57; 3.2.58; 6.3.2.2.2;  
6.3.4.4.1; 7.3.2.4.

**Device-specific parameter:** 3.2.12; 3.2.14; 3.2.28;  
6.2.3.4; 6.3.8.2; 8.1.

**Durability:** 3.2.13; 6.2.6.2; 7.1.2; 7.3.1.

**Dynamic module of legally relevant software:**  
3.2.14; 6.2.3.1; 6.3.2.2.1; 6.3.2.2.5; 6.3.4.2; 6.3.5.2;  
6.3.6.8; 6.3.6.9; 6.3.7; 7.1.1; 7.1.2; 7.3.2.1; 7.3.2.2;  
7.3.2.5; 8.1;

**Electronic measuring instrument:** 3.2.15; 3.2.22.

**Electronic Signature:** 3.2.16; 6.2.3.5; 6.3.4.3;  
6.3.5.3; 6.3.8.4.5.

**Error (of indication):** 3.2.17; 3.2.22.

**Error log:** 3.2.18; 6.2.6.1; 6.2.6.2.

**Evaluation (software):** 7.1.1; 7.1.2; 7.2.1; 7.3.1;  
7.3.2.1; 7.3.2.2; 7.3.2.3; 7.4.

**Evaluation (type):** 3.2.65; 3.2.70; 6.2.3.4; 6.2.6.1;  
6.2.6.2; 6.3.2.2.2; 6.3.7; 7.1.1; 7.1.2; 7.2.1;

**Event:** 3.2.1; 3.2.19; 3.2.20; 3.2.58; 3.2.63; 6.2.3.5;  
6.2.3.6; 6.3.8.4.4; 6.3.8.4.8; 6.3.9.3.3; 7.2.2; 8.3.2; 8.3.3.2.

**Event counter:** 3.2.20; 6.2.3.5; 6.2.3.6; 6.3.2.1.3;  
6.3.8.4.4; 7.2.2.

**Executable code:** 3.2.21; 3.2.60; 6.2.1.

**Fault:** 3.2.22; 3.2.51; 7.1.2; 7.3.2.3.

**Hash function:** 3.2.23; 6.2.6.1.

**Integrity (of programs, data, or parameters):**  
3.2.10; 3.2.16; 3.2.24; 3.2.53; 3.2.62; 6.3.2.1.4;  
6.3.2.1.7; 6.3.4.3; 6.3.5.3; 6.3.6.3.2; 6.3.6.3.3; 6.3.8.4.1;  
6.3.8.4.6; 6.3.8.4.7; 6.3.9.1.2; 6.3.9.1.8; 6.3.9.3.1;  
6.3.9.3.3; 6.3.9.4; 7.2.2; 8.1; 8.2.2; 8.2.3.2; 8.3.1;  
8.3.2; 8.3.3.1; 8.3.3.2; 8.3.4.1; 8.3.5; 8.3.5.1.

**Interface:** 3.2.7; 3.2.25; 3.2.47; 3.2.56; 3.2.58; 3.2.59;  
3.2.68; 6.2.1; 6.2.3.1; 6.2.3.2; 6.2.3.3; 6.2.4; 6.3.2;  
6.3.2.1.1; 6.3.2.1.2; 6.3.2.1.7; 6.3.2.2.2; 6.3.2.2.3;  
6.3.2.2.4; 6.3.3; 6.3.6.1; 6.3.6.2; 6.3.6.3.4; 6.3.6.3.5;  
6.3.6.6; 6.3.6.7.1; 6.3.9.1.9; 6.3.9.1.10; 7.1.1;  
7.1.2; 7.3.1 7.3.2.1; 7.3.2.3; 7.3.2.4; 8.3.1.

**Legally relevant:** 2.1; 3.2.1; 3.2.12; 3.2.14; 3.2.18,  
3.2.19; 3.2.27; 3.2.28; 3.2.29; 3.2.39; 3.2.47; 3.2.52;  
3.2.53; 3.2.56; 3.2.59; 3.2.66; 3.2.67; 4.5; 4.7; 6.2.1;  
6.2.3.1; 6.2.3.2; 6.2.3.3; 6.2.3.4; 6.2.3.5; 6.2.3.6; 6.2.6.1;  
6.2.7; 6.3.2; 6.3.2.1.1; 6.3.2.1.2; 6.3.2.1.3; 6.3.2.1.4;  
6.3.2.1.6; 6.3.2.1.7; 6.3.2.2.1; 6.3.2.2.2; 6.3.2.2.3;  
6.3.2.2.4; 6.3.2.2.5; 6.3.3; 6.3.4.2; 6.3.4.3; 6.3.4.4.1;  
6.3.5.2; 6.3.5.3; 6.3.6.1; 6.3.6.2; 6.3.6.3.1; 6.3.6.3.2;  
6.3.6.3.4; 6.3.6.4; 6.3.6.5.1; 6.3.6.5.2; 6.3.6.5.4; 6.3.6.6;  
6.3.6.7.1; 6.3.6.7.2; 6.3.6.8; 6.3.6.9; 6.3.7; 6.3.8.1;  
6.3.8.2; 6.3.8.3; 6.3.8.4.1; 6.3.8.4.4; 6.3.8.4.5; 6.3.8.4.8;  
6.3.9.1.1; 6.3.9.1.2; 6.3.9.1.3; 6.3.9.1.9; 6.3.9.3.3; 7.1.1;  
7.1.2; 7.3.2.1; 7.3.2.2; 7.3.2.5; 8.1; 8.3.2; 8.3.3.2; 8.3.4.2.

**Legally relevant parameter:** 3.2.12; 3.2.19; 3.2.28; 3.2.66; 6.2.1; 6.2.3.1; 6.2.3.4; 6.2.3.5; 6.2.6.1; 6.3.8.4.1.

**Legally relevant software:** 2.1; 3.2.14; 3.2.19; 3.2.29; 3.2.39; 3.2.47; 3.2.47; 3.2.52; 3.2.53; 3.2.59; 3.2.66; 4.7; 6.2.3.1; 6.2.3.2; 6.2.3.4; 6.2.3.6; 6.2.4; 6.2.6.1; 6.2.7; 6.3.2; 6.3.2.1.2; 6.3.2.1.6; 6.3.2.2.1; 6.3.2.2.2; 6.3.2.2.3; 6.3.2.2.4; 6.3.2.2.5; 6.3.3; 6.3.4.2; 6.3.4.3; 6.3.5.2; 6.3.5.3; 6.3.6.2; 6.3.6.3.1; 6.3.6.3.2; 6.3.6.4; 6.3.6.5.2; 6.3.6.5.4; 6.3.6.6; 6.3.6.8; 6.3.6.9; 6.3.7; 6.3.8.1; 6.3.8.2; 6.3.8.3; 6.3.8.4.4; 6.3.8.4.5; 6.3.8.4.8; 6.3.9.1.1; 6.3.9.1.2; 6.3.9.1.3; 7.1.1; 7.1.2; 7.3.2.1; 7.3.2.2; 7.3.2.5; 8.1; 8.3.2; 8.3.3.2; 8.3.4.2.

**Maximum permissible error:** 3.2.30; 3.3; 7.3.2.2.

**Measuring instrument:** 1; 2.1; 2.2; 2.3; 3.1; 3.2.1; 3.2.2; 3.2.5; 3.2.7; 3.2.8; 3.2.9; 3.2.12; 3.2.13; 3.2.15; 3.2.18; 3.2.19; 3.2.21; 3.2.22; 3.2.26; 3.2.27; 3.2.28; 3.2.29; 3.2.30; 3.2.31; 3.2.41; 3.2.42; 3.2.46; 3.2.48; 3.2.49; 3.2.51; 3.2.53; 3.2.55; 3.2.57; 3.2.58; 3.2.59; 3.2.65; 3.2.66; 3.2.68; 3.2.70; 4.3; 4.4; 5.1; 5.2; 6.1; 6.2; 6.2.1; 6.2.2; 6.2.3.1; 6.2.3.2; 6.2.3.3; 6.2.3.4; 6.2.3.5; 6.2.4; 6.2.5; 6.2.6.1; 6.2.6.2; 6.2.7; 6.3.1; 6.3.2; 6.3.2.1.1; 6.3.2.1.3; 6.3.2.1.6; 6.3.2.1.7; 6.3.2.1.8; 6.3.2.2.1; 6.3.2.2.2; 6.3.2.2.4; 6.3.3; 6.3.4.2; 6.3.4.3; 6.3.4.4.1; 6.3.5.2; 6.3.5.3; 6.3.5.4; 6.3.6.1; 6.3.6.3.4; 6.3.6.7.1; 6.3.7; 6.3.8.1; 6.3.8.2; 6.3.8.3; 6.3.8.4.1; 6.3.8.4.2; 6.3.8.4.3; 6.3.8.4.4; 6.3.8.4.5; 6.3.8.4.7; 6.3.8.4.8; 6.3.8.4.9; 6.3.9.1; 6.3.9.1.2; 6.3.9.1.3; 6.3.9.1.4; 6.3.9.1.5; 6.3.9.1.8; 6.3.9.1.11; 6.3.9.1.12; 6.3.9.2; 6.3.9.3.2; 6.3.9.3.3; 7.1.1; 7.1.2; 7.2.1; 7.2.2; 7.3.2.1; 7.3.2.2; 7.3.2.3; 7.5; 8.1; 8.3.1; 8.3.2; 8.3.3.1; 8.3.3.2; 8.3.4.1; 8.3.5.1; 8.3.6.1; 8.3.6.2; 8.3.6.3.

**Mobile app:** 3.2.45;

**Non-interruptible/interruptible measurement:** 3.2.26; 3.2.46; 6.2.6.1.

**Operating system:** 3.2.4; 6.2.3.1; 6.3.2.1.7; 6.3.2.2.1; 6.3.2.2.3; 6.3.2.2.4; 6.3.3; 6.3.6.1; 6.3.6.2; 6.3.6.3.4; 6.3.6.4; 6.3.6.5.2; 6.3.6.5.4; 6.3.6.6; 6.3.6.7.1; 6.3.6.7.2; 6.3.6.8; 6.3.6.9; 7.1.2.

**Performance:** 3.2.8; 3.2.13; 7.1.1; 7.2.1; 8.3.6.3.

**Program code:** 3.2.56; 6.2.6.1; 6.3.2.2.2; 6.3.4.3; 6.3.5.3; 8.2.2.

**Protective interface:** 3.2.47; 6.2.3.2; 6.2.3.3; 6.2.4; 6.3.2.1.1; 6.3.2.1.2; 6.3.2.1.7; 6.3.2.2.2; 6.3.2.2.3; 6.3.6.2; 6.3.6.1; 6.3.6.6.

**Remote verification:** 3.2.48; 3.2.62; 6.2.1; 6.2.3.3; 6.2.3.6; 6.2.4; 6.2.6.3; 6.3.4.1; 6.3.5.1; 6.3.9; 6.3.9.1;

6.3.9.1.1; 6.3.9.1.4; 6.3.9.1.5; 6.3.9.1.6; 6.3.9.1.8; 6.3.9.1.9; 6.3.9.1.10; 6.3.9.1.11; 6.3.9.1.12; 6.3.9.2; 6.3.9.3.1; 6.3.9.3.3; 7.1.2; 7.2.2; 8.3; 8.3.1; 8.3.2; 8.3.3.1; 8.3.3.2; 8.3.4.1; 8.3.4.2; 8.3.5.1; 8.3.6.1; 8.3.6.2; 8.3.6.3; 8.3.6.4; 8.3.6.5.

**Sealing:** 3.2.54; 5.2; 6.2.3.1; 6.2.3.5.

**Securing:** 3.2.16; 3.2.27; 3.2.50; 6.3.2.1.1; 6.3.2.1.2; 6.3.2.1.7; 6.3.3; 6.3.8.3; 6.3.8.4.3; 6.3.9.1.12; 7.2.2; 8.1.

**Software examination:** 3.2.54; 6.2.2; 7.2.1.

**Software identification:** 3.2.55; 6.2.1; 6.3.4.1; 6.3.5.1; 6.3.8.4.8; 6.3.9.3.3; 7.1.2; 7.2.2; 7.3.2.3; 8.1; 8.3.2; 8.3.3.2.

**Software interface:** 3.2.56; 3.2.59; 6.3.2.2.3; 7.1.1; 7.1.2; 7.3.2.4.

**Software module:** 3.2.7; 3.2.11; 3.2.14; 3.2.19; 3.2.28; 3.2.29; 3.2.47; 3.2.52; 3.2.55; 3.2.56; 3.2.57; 3.2.59; 3.2.62; 3.2.66; 3.2.68; 6.2.1; 6.2.3.1; 6.2.3.2; 6.2.6.1; 6.3.2; 6.3.2.1.1; 6.3.2.1.6; 6.3.2.2.1; 6.3.2.2.2; 6.3.2.2.3; 6.3.4.3; 6.3.5.3; 6.3.6.2; 6.3.6.6; 6.3.6.7.2; 6.3.8.4.5; 6.3.9.1.1; 7.1.2; 7.2.2; 7.3.2.2; 7.3.2.3; 7.3.2.4; 7.3.2.5; 7.3.2.6; 7.5; 8.3.2.

**Software protection:** 3.2.58; 6.2.3.1; 6.2.3.5; 7.3.2.3.

**Software separation:** 3.2.59; 6.2.1; 6.3.2.2.1; 6.3.2.2.2; 6.3.2.2.4; 7.3.2.4.

**Source code:** 3.2.60; 7.1.2; 7.3.2.2; 7.3.2.4; 7.3.2.5; 7.3.2.6.

**Storage device:** 3.2.61; 6.3.2.1.7; 6.3.4.4.1; 6.3.8.4.8.

**Test:** 5.1; 6.2.2; 6.2.7; 6.3.8.4.1; 6.3.8.4.7; 6.3.9.3.2; 6.3.9.3.3; 7.1.1; 7.2.1; 7.3.2.1; 7.3.2.2; 7.3.2.3; 7.3.2.6; 7.4; 7.5; 8.1; 8.2; 8.3.4.1; 8.3.6.

**Time stamp:** 3.2.1; 3.2.58; 3.2.63; 6.2.3.6; 6.2.7; 6.3.2.1.2; 6.3.4.2; 6.3.5.2; 6.3.6.7.2; 6.3.8.4.8; 6.3.9.1.5.

**Transmission of measurement data:** 3.2.64; 6.2.3.1; 6.3.2.1.1; 6.3.2.1.7; 6.3.5.1; 6.3.5.2; 6.3.5.3; 6.3.5.4.

**Type-specific parameter:** 3.2.28; 3.2.66; 6.2.3.4.

**Type evaluation authority:** 6.2.3.2; 6.2.3.3; 6.3.2.1.1; 6.3.2.2.2; 7.1.2.

**Universal device:** 3.2.67; 5.2; 6.2.3.1; 6.3.2.1.1; 6.3.2.1.6; 6.3.2.2.3; 6.3.2.2.4; 6.3.6.9.

**User interface:** 3.2.68; 6.2.1; 6.2.3.2; 6.3.3; 7.1.2; 7.3.2.3.

**Verification:** 3.2.48; 3.2.62; 3.2.69; 3.2.70; 6.2.1;  
6.2.3.3; 6.2.3.4; 6.2.3.6; 6.2.6.3; 6.3.4.1; 6.3.4.3;  
6.3.5.1; 6.3.6.7.1; 6.3.8.1; 6.3.8.2; 6.3.8.3; 6.3.8.4.1;  
6.3.8.4.8; 6.3.8.4.9; 6.3.9; 6.3.9.1; 6.3.9.1.1; 6.3.9.1.4;  
6.3.9.1.5; 6.3.9.1.6; 6.3.9.1.8; 6.3.9.1.9; 6.3.9.1.10;  
6.3.9.1.11; 6.3.9.1.12; 6.3.9.2; 6.3.9.3.1; 6.3.9.3.3;  
7.1.1; 7.1.2; 7.2.1; 7.2.2; 7.3.2.2; 7.3.2.3; 7.3.2.6; 7.4;  
8.1; 8.2; 8.2.1; 8.2.3.1; 8.3; 8.3.1; 8.3.2; 8.3.3.1;  
8.3.3.2; 8.3.4.1; 8.3.4.2; 8.3.5; 8.3.5.1; 8.3.6.1;  
8.3.6.2; 8.3.6.3; 8.3.6.4; 8.3.6.5.