D 31 – 2.1CD

CLEAN VERSION

Title: General requirements for software-controlled measuring instruments

Contents

Foreword **5**

1 Introduction **7**

2 Scope and field of application **7**

3 Terms and definitions **8**

3.1 General 8

3.2 General terminology 8

3.3 Abbreviations 20

4 Instructions for use of this Document in drafting OIML Recommendations **21**

5 Risk assessment **22**

6 Requirements for measuring instruments with respect to the software **23**

6.1 General 23

6.2 General requirements 23

6.2.1 Conformity of manufactured devices to the approved type 23

6.2.2 Functional requirements 23

6.2.3 Securing and protection 27

6.3 Requirements specific for configurations 32

6.3.1 General 32

6.3.2 Detection of significant defects 32

6.3.3 Detection of durability errors and/or significant faults 33

6.3.4 Dynamic modules of legally relevant software 34

6.3.5 Compatibility of operating systems and hardware 35

6.3.6 Data storage 38

6.3.7 Data transmission 41

6.3.8 Specification and separation of legally relevant components and software modules 43

6.3.9 Maintenance and reconfiguration 48

6.3.10 Remote verification capability 53

7 Type evaluation **56**

7.1 Software documentation to be supplied for type evaluation 56

7.2 Requirements for the evaluation procedure 58

7.3 Verification and evaluation methods 59

7.4 Software evaluation procedure 70

7.5 Equipment under test (EUT) 70

8 Verification of a measuring instrument 71

8.1 General 71

8.2 Verification methods, test items 71

8.3 Remote verification 72

Annex A Bibliography (Informative) 77

Annex B Example of a software test report (Informative) 80

Annex C Remarks on measurement terminology (Informative) 100

Annex D How to adopt requirements in an OIML Recommendation (Informative) . 103

Annex E Comparison table 109

Annex F Index 111

Foreword

The International Organization of Legal Metrology (OIML) is a worldwide, intergovernmental organization whose primary aim is to harmonize the regulations and metrological controls applied by the national metrological services, or related organizations, 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 harmonize 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 20xx (E) – was developed by Project Group 4 of OIML Technical Subcommittee TC 5/SC 2 *Software*. It was approved for final publication by the International Committee of Legal Metrology at its 58th meeting in 20xx and will be submitted to the International Conference on Legal Metrology in 20yy 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 Organization'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
Internet: 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 and Corresponding Members in the implementation of OIML Recommendations in their national legislation.

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 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 legislation for this area needs 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 1:2022 [2]), with the OIML International Document *General requirements for measuring instruments – Environmental conditions* (OIML D 11:2013 [3]) 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 certificate.

3.2 General terminology

References to external sources are provided for the definitions themselves, notes have only been copied or adapted where relevant.

3.2.1 audit trail

continuous data containing a timestamped 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 3.2.20.

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 a 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.).

3.2.6 cloud

servers that are accessed over the internet or another network, and the software and databases that run on those servers

Note: Cloud servers may not be physically accessible to all parties and may be located in a different country. Their physical location may not be known and not fixed.

3.2.7 communication interface

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

Note 1: Communication interfaces can utilize wired, optical, radio, etc. communication 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 hardware part of a measuring instrument that performs a specific function or functions

Note: Components can be part of or identical to modules as defined in V1 4.04.

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, plus a trusted party digital signature

Note: The trusted party digital signature binds the public key to the unique identification of the subject.

3.2.10 cryptographic means

means such as encryption and decryption with the purpose of providing confidentiality, or hashes and digital signatures (see 3.2.14) 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 measuring instrument, component, and/or software 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.

3.2.13 digital data processing unit

part of a measuring instrument which only receives digital input data and generates digital output data

3.2.14 digital signature

software means which is added to software or data with the purpose of verifying 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 digital 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.

Note 4: A digital signature provides nonrepudiation: the signee cannot deny signing the software or data.

3.2.15 durability

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

[OIML V 1:2022, 5.15]

3.2.16 dynamic module of legally relevant software

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

Note 1: Such dynamic modules incorporate or utilize machine learning or artificial intelligence characteristics and processes.

Note 2: This includes software modules that can have an influence on legally relevant software.

3.2.17 electronic measuring instrument

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

[OIML D 11:2013, 3.1]

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

3.2.18 error of indication

indication minus a reference quantity value

[OIML V 1:2022, 0.04]

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.

3.2.19 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

3.2.20 event

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

[OIML V 1:2022, 6.06]

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 or characteristics.

3.2.21 event counter

non-resettable counter that increments each time an event occurs

3.2.22 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.

3.2.23 fault

difference between the error of indication and the intrinsic error of a measuring instrument

[OIML V 1:2022, 5.12]

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.

3.2.24 hash function

a (mathematical) function which maps data of arbitrary size into data of a fixed size called a digest

[ISO/IEC 9594-8:2020] [4]

Note 1: 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.

Note 2: A cryptographic hash function has three additional properties: collision resistance, preimage resistance, and second preimage resistance, where preimage resistance refers to the inability (computational infeasibility) to reconstruct a preimage or message from a message digest.

3.2.25 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

Note: Software may include parameters and data, see 3.2.73.

3.2.26 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 2382-9:2015] [5]

3.2.27 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 totalizing automatic weighing instrument, b) fuel dispenser.

Note 2: See also non-interruptible cumulative measurement (3.2.49).

3.2.28 intrinsic error

error of indication, determined under reference conditions

[OIML V 1:2022, 0.06]

3.2.29 learning facility

aspect of a dynamic module of legally relevant software that adapts device-specific parameters over time

3.2.30 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 (e.g., the metrological functions, securing and protection features), and/or critical for the completion of the transaction, are also under legal control.

Note 2: The relevant Recommendations define what is legally relevant and formulate requirements to those items (e.g., data, functions, securing and protection features and information for the completion of the transaction).

Note 3: Any property of the measuring instrument not subject to legal control is referred to as legally non-relevant in this Document, see usage of the term in OIML V1:2022 6.02.

3.2.31 legally relevant parameter

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

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

3.2.32 legally relevant software

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

3.2.33 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

[OIML V 1:2022, 0.05]

3.2.34 measuring instrument

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

[OIML V 1:2022, 0.10]

3.2.35 measurement

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

[OIML V 2-200:2012, 2.1]

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.

3.2.36 measurement data

data used during the measurement process

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

3.2.37 measurement error

measured quantity value minus a reference quantity value

[OIML V 2-200:2012, 2.16]

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.

Note 3: See Annex C for clarification regarding measurement-related terms.

3.2.38 measurement metadata

metadata related to the measurement process

Note: Measurement metadata include the measured quantity value metadata, measurement result relevant metadata and measurement process metadata, see Annex C.

3.2.39 measurement process data

data used during the measurement process to construct the measurement result

Note 1: Examples of measurement process data include values of measurement parameters, values of connection settings or values of session parameters.

Note 2: See Annex C for clarification regarding measurement-related terms.

3.2.40 measurement process information

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

Note: Measurement process information include measurement process data and measurement process metadata, see Annex C.

3.2.41 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, see Annex C.

3.2.42 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 measurements, 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.

Note 3: See Annex C for clarification regarding measurement-related terms.

3.2.43 measured quantity value metadata

metadata related to the measured quantity value

Note: See Annex C for clarification regarding measurement-related terms.

3.2.44 measurement result relevant data

data used during the process of constructing the measurement result

Note: Examples of measurement result relevant data include digital numbers or analog values originating from a sensor or measuring instrument ID. In cases where it is part of the measurement result, see Annex C.

3.2.45 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 analog 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, see Annex C.

3.2.46 measurement result relevant information

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

Note: Measurement result relevant information include measurement result relevant data and measurement result relevant metadata, see Annex C.

3.2.47 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]

3.2.48 mobile app

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

[Cambridge Dictionary, fourth edition, 2021]

3.2.49 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 totalizing automatic weighing instrument, b) heat meter.

Note 2: See also interruptible cumulative measurement (3.2.27).

3.2.50 OIML certificate

type examination certificate, issued by an OIML Issuing Authority, attesting the conformity of a type of a measuring instrument or module with the relevant requirements of an OIML Recommendation at the time of testing and evaluation

[OIML B 18:2025, 3.26]

3.2.51 operating system

software to control program operation and to provide services for resource allocation, task scheduling, I/O control, and data management as well as tasks like access control and security

3.2.52 protection

means making an intervention impossible or evident

3.2.53 protective interface

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

Note: The protective interface consists of program code and dedicated data domains. Defined commands or data are exchanged between the software modules by having one part of the protective interface store them in the dedicated data domain, while the other part of the protective interface reads them from it. Writing and reading code is part of the protective interface.

3.2.54 remote verification

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

3.2.55 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.

3.2.56 securing

means preventing unauthorized access to hardware or software

[OIML V 1:2022, 2.21]

Note: This may be achieved by means of passwords.

3.2.57 significant defect

incident that has an undesirable impact on the compliance of the measuring instrument with requirements of this Document

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

3.2.58 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 a software module; 3) evolved parameters of the dynamic parts of the module

3.2.59 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.

3.2.60 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)

3.2.61 software identification

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

Note 1: Software identification can be checked on an instrument whilst in use, see 6.2.2.1.

Note 2: Software identifiers are individual instances of the software identification.

3.2.62 software interface

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

[OIML V 1:2022, 6.03]

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.

3.2.63 software module

software entity such as a program, subroutine, library, parameter or dataset, 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.

3.2.64 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 3.2.21, or a data file containing timestamped information records the event.
3) The interface of a measuring instrument is physically protected by means of a hardware seal, so that accessing that interface can only be achieved by breaking, removing or damaging the seal.

Note: See 6.2.3.1.

3.2.65 software separation

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

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

3.2.66 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.

3.2.67 storage device

internal or external device used for storing measurement data that are necessary to reconstruct the measurement result

Note: See Annex C for clarification regarding measurement-related terms.

3.2.68 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 as part of remote verification procedures.

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

3.2.69 timestamp

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

3.2.70 traced update

procedure of changing software in a measuring instrument or component after which a subsequent verification is not necessary.

3.2.71 transmission of measurement data

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

3.2.72 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 and/or an evaluation certificate

[OIML V 1:2022, 2.04]

3.2.73 type-specific parameter

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

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

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.

3.2.74 universal device

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

Example:

1) An electricity meter without a display communicates over a network with a tablet that is configured by software to act as the indication device.
2) An industrial PC is used for the pre-processing of multidimensional measurement data. The data is transmitted from multiple sensors to indication via an Ethernet subnetwork.

3.2.75 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, and software window on a screen, including the software to generate it.

3.2.76 verification

provision of objective evidence that a given item fulfils specified requirements

[V 2-200:2012, 2.44]

3.2.77 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

[OIML V 1:2022, 2.09]

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

3.2.78 verification software

software on a remote unit used for the purpose of verification of a measuring instrument

3.2.79 verified update

procedure of changing software in a measuring instrument or component after which a subsequent verification is necessary

3.3 Abbreviations

CRC	Cyclic Redundancy Check
EUT	Equipment Under Test
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Organization for Standardization
IT	Information Technology
MPD	Measurement Process Data
MPE	Maximum Permissible Error
MPM	Measurement Process Metadata
MQV	Measured Quantity Value
MQVM	Measured Quantity Value Metadata
MRRD	Measurement Result Relevant Data
MRRI	Measurement Result Relevant Information
MRRM	Measurement Result Relevant Metadata
OIML	International Organization of Legal Metrology
PG	Project Group
SW	Software

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.The provisions of this Document may also be used by OIML PGs in the development and revision of OIML Guides where relevant.

4.2 Annex D provides a detailed overview of the necessary steps PGs should take when adopting this Document. To facilitate the implementation, guidance for PGs, documentation requirements and information to be contained in OIML certificates are marked as such. The following different categories of guidance for PGs are used in this Document:

- Notes – informative notes to aid understanding and interpretation of the clause
- Guidance – guidance to PGs regarding sotware-related requirements to be considered within a Recommendation
- Documentation – guidance to PGs regarding documentation to be submitted to support software evaluation in accordance with a Recommendation
- Certificate – guidance to PGs regarding software-related information to be included in a type evaluation certificate issued in accordance with a Recommendation

The guidance for PGs uses the normative verbs “may” and “should”. “May” signifies that guidance is optional and the requirement can stand on its own. “Should” implies that PGs have to follow the guidance because the requirement is incomplete otherwise.

4.3 It is the objective of this Document to provide the PGs responsible for drawing up OIML Recommendations and OIML Documents 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, specifically regarding securing and protection of the metrological characteristics.

Guidance: PGs should determine which risk level is suitable. In clause 5, some aid is given for performing this task.

Guidance: PGs should decide which metrological characteristics (at least legally relevant software, parameters and measurement data) shall comply with the requirements laid out in the following clauses.

4.4 *Guidance:* PGs should decide which parameters are legally relevant for a specific application.

Guidance: PGs should decide which measurement data are legally relevant and shall comply with the requirements, see Annex C. PGs should also decide which metadata shall be documented by the manufacturer.

Note: All referenced 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 referenced documents.

5 Risk assessment

5.1 This clause is intended as a guide to determine a set of risk levels to be generally applied for acceptable technical solutions and 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) possibility to repeat a measurement or to interrupt it.

f) possibility to check the measurement at a later point.

PGs should consider risk assessment standards when deciding risk levels, e.g., ISO/IEC 27005 [6].

The level of examination and the risk level are linked. If a raised risk level is applied and unless a hardware seal is used, e.g., on open-wired communication interfaces or the housing, an extended examination of the software to detect deficiencies or security vulnerabilities should be performed.

6 Requirements for measuring instruments regarding software

6.1 General

Note 1: The requirements are separated into:

- general requirements (6.2). The general requirements represent the state of the art in information technology (IT) at the time of publication. In principle, they are applicable to all kinds of software-controlled measuring instruments and components of measuring instruments. They should be considered in all Recommendations.
- requirements for specific configurations (6.3). The specific configurations cover additional requirements for technical features that are only mandatory in select Recommendations or added as a feature by the manufacturer.

Note 2: 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 the case of normal risk level;

(II) Technical solution acceptable in the case of raised risk level (see clause 5).

6.2 General requirements

6.2.1 Conformity of manufactured devices to the approved type

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

Note 1: OIML D 34:2019 [7] interprets certification as consisting of type evaluation and type approval.

Note 2: 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 the presence of dynamic parameter changes, see 6.3.4.

6.2.2 Functional requirements

6.2.2.1 Software identification

Software modules of a measuring instrument or component shall be unambiguously, uniquely and correctly identified.

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

The software identification (see 3.2.61) linked to the software may consist of more than one part, see also software separation (3.2.65 and 6.3.8.3.2), but at least one part shall be dedicated to the legal purpose.

Guidance: PGs may decide which forms of software identification are permissible.

Documentation: The documentation shall contain a description of the software identification and instructions on how to display or print the software identification.

Certificate: 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 given in the certificate.

The software 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, the software identification shall be sent via a communication interface in order to be displayed or printed on another legally relevant component.

If the instrument facilitates remote verification, the software identification shall also be sent to the verification software.

As an exception, the software identification may be marked on the instrument or component concerned 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.

d) The software identification is correctly marked on the instrument or component concerned.

Guidance: PGs should allow or disallow this exception.

Regardless of the form of the software identification, it shall be readily available when the instrument is in service to allow it to be checked, see 6.2.1 and 6.2.2.7.

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. The version number of a flow meter has the following structure: A.Y.Z., where A represents the version of the core software that is counting pulses; Y represents the version of the conversion function (none, at 15 °C, at 20 °C); Z represents the language of the user interface.

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

6.2.2.2 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, displaying or printing measurement results, etc.).

Means shall be taken to minimize the risk of violating the MPE if a learning facility is used for dynamic modules of legally relevant software.

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

Documentation: The documentation shall contain a list of all legally relevant functions and a description of how to access them. There shall be no hidden or undocumented legally relevant functions. Additionally, the documentation shall contain a description of the accuracy of the measurement algorithm.

6.2.2.3 Prevention of misuse

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 and that the possibilities for accidental, unintentional or intentional misuse are minimal.

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

Examples:

1) (I)/(II) The user is guided by menus. The legally relevant functions are combined into one branch in this menu. If legally relevant parameters are about to be changed by an action, the user is warned and requested to make a confirmation before the function is executed. See also 6.2.3.4.

2) (I)/(II) The measurement is started remotely by a mobile app, which runs on an arbitrary device. The measuring instrument itself is fully secured and protected by hardware and software means. 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 mobile device, such as a smartphone, for indication.

6.2.2.4 Indications

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

The measurement result (measured quantity value and measurement result relevant data) shall be displayed or printed correctly and accompanied by all measurement result relevant data necessary to inform the user of the significance of the result.

Guidance: PGs should specify the measurement result relevant data that need to be indicated.

Guidance: PGs should specify the layout of the display and printout for the legally relevant information.

Guidance: PGs may specify the requirements for the display and/or printout of the legally relevant information.

6.2.2.5 Shared indications

A display or printout may be employed to present both information from the legally relevant software and other information.

If a display or printout is used both for legally relevant and legally non-relevant information, the legally relevant information shall always be readable, and clearly distinguishable from legally non-relevant information.

Examples:

1) (I) In a measuring instrument that realizes software separation, the measurement results are displayed in a separate software window. The means described in 6.3.8.3.3 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-window user interface. The window displaying the legally relevant data is generated and controlled by procedures in the legally relevant dynamically linkable library (see 6.3.8.3). 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 a measuring instrument that realizes software separation, the measurement application runs in kiosk mode. This mode is a feature that limits a device to running specific applications and settings and does not allow the user to start other applications. The entire display is controlled by the legally relevant software. Legally non-relevant data are presented in a separated part of the display marked as legally non-relevant.

3) (II) A mobile app on a device belonging to the measuring instrument is used to indicate measurement results calculated on a separate component. Since the mobile device is also used for other, legally non-relevant, purposes, the operating system of the mobile device is configured in accordance with 0. Whenever the legally relevant mobile app is running, the user is informed accordingly by the app. To ensure that the measurement result can always be distinguished from legally non-relevant information, legally relevant measurement data are only made available to legally non-relevant mobile apps after primary indication on the legally relevant mobile app.

4) (II) The operating system ensures that indicated legally relevant information is distinguishable from other legally non-relevant information." Any other technical change would not be possible at this stage.

6.2.2.6 Timestamps

Note: Timestamps (see 3.2.69) are typically used to record when a particular event occurred, or as measurement result relevant data to specify when a measurement took place.

The use of timestamps is mandatory if audit trails are used.

If a timestamp is required for the legally relevant purpose, the instrument shall be able to keep or read time accurately, whether via an internal clock or an external clock synchronized with legal time.

Note: If setting the clock or clock accuracy are legally relevant, especially in the case of an external clock, see 6.2.3.5.

Guidance: PGs may define minimum requirements for the resolution of timestamps to ensure consistency and applicability.

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

Example:

(II) As an internal clock, a measuring instrument uses a second counter represented in the "date and time" format in accordance with ISO 8601 [8].

6.2.2.7 Information for verification

It shall be possible to display, print, and/or transmit to the verification software all necessary verification information, see 6.3.10.

Necessary verification information may include:

a) the software identification;

b) current legally relevant parameter settings;

c) data containing evidence of an intervention.

Guidance: PGs may define what verification information is necessary for the instrument type.

Note: Audit trail(s) or event counter(s) are a means to provide evidence of an intervention, see also 6.2.3.3.

Certificate: The certificate shall describe how this information can be displayed or printed and specify how it can be obtained by the remote verification procedure.

6.2.2.8 Contents of the audit trail

The audit trail shall contain, at a minimum, the following information:

- timestamp of the event;
- in the case of a parameter change:
  - identification of the changed parameter;
  - the old and new value of the changed parameter;
- in the case of a traced update, see 0;

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

Documentation: The documentation shall contain a description of how to access the audit trail.

Guidance: PGs may define additional information to be recorded in the audit trail. This could be the source of the modification, details of a verified update or an indication that the modification is the result of an action of dynamic modules of legally relevant software or of remote verification.

6.2.3 Securing and protection

6.2.3.1 General

A measuring instrument shall be provided with the means to protect its metrological properties.

Software protection shall comprise appropriate sealing by hardware or software means, making an intervention impossible or evident.

In the case of a software seal, a checking facility shall check if no changes have occurred; if the check fails, this is considered a significant defect (see 6.3.2).

Documentation: The documentation shall contain a description of the means to validate the conformity of devices in use, a list of commands of each hardware interface of the measuring instrument or component, an overview of the system hardware (topology block diagram, type of computer(s), type of network, etc.), a description of the user interface, menus and dialogs, and the operating manual.

6.2.3.2 Software

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

Note 1: Accidental changes include changes due to physical effects. Unintentional changes include a user mistakenly resetting parameters to factory settings. Intentional changes include modification of the software, loading different software modules, or changing software by swapping the memory device that contains the software, or unauthorized updates.

Note 2: Downloading software into the measuring instrument or component is allowed if the requirements for updates are fulfilled, see 6.3.9.3 and 6.3.9.4.

Examples:

1) (I) A measuring instrument consists of two components, one containing the main metrological functions incorporated in a sealed housing. 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 1) in 6.3.8.3.3).

2) (I)/(II) The housing containing the memory device with the software is protected by a hardware seal, or the memory device is fixed on the printed circuit board by means of a hardware seal to prevent swapping the memory device.

3) (I)/(II) To prevent software on a memory device from being changed, the write-enable input of the memory device that contains the software is inhibited by a switch that can be protected by a hardware seal. The circuit is designed in such a way that the write protection cannot be cancelled by a short-circuit of contacts.

4) (II) *Cryptographic means*. A cryptographic certificate may be used. The software is signed by a trustworthy institution (e.g., an OIML issuing authority) with a digital signature. The authenticity of the signed software can be verified by using the public key of the trustworthy institution and decrypting the digital signature of the certificate. The measuring instrument itself regularly checks the digital signature. If the check fails, an error is recorded in an audit trail and all further measurements are inhibited.

6.2.3.3 Means to provide evidence of an intervention

Note 1: Audit trail(s) and event counter(s) are specific examples of 'means to provide evidence of an intervention' (see also 6.2.2.7).

Note 2: Regarding the content of the audit trail, see 6.2.2.8

Audit trail(s) and event counter(s) are part of the legally relevant software and shall be secured and protected as such against accidental, unintentional or intentional changes.

The reference number of an event counter shall be fixed and protected by appropriate hardware means at the time of (initial or subsequent) verification. This reference number shall be visibly marked on the instrument.

It shall not be possible to change the data of an event counter or of an entry in an audit trail, even in the event of a software update.

It shall not be possible to delete an event counter or an entry in an audit trail, except as an appropriate response to free up storage capacity to allow for the addition of new entries.

Any change to the recorded data in event counter(s) or audit trail(s), except for deletions to free up storage capacity (as specified above), is a significant software defect and shall be handled accordingly (see detection of significant defects, 6.3.2).

Events shall be recorded automatically, unless PGs determine specific provisions to apply.

Guidance: For specific types of instruments, PGs should define which manual additions to an event recorded in the audit trail(s) are admissible. Further, these additios should only be admissible if they do not affect the remaining contents of the audit trail(s).

The audit trail(s) and event counter(s) shall have sufficient capacity to ensure the traceability of events between at least two successive verifications or inspections of a measuring instrument in the field.

Note: This requirement enables inspection authorities that are responsible for the metrological surveillance of legally controlled measuring instruments to back-trace events over an adequate period of time (depending on national legislation).

If an audit trail or event counter has no more capacity, an appropriate response is required.

Guidance: PGs should specify a sufficient capacity required for the audit trail(s) and event counter(s) and the response required, i.e., either the oldest entry may be deleted, or no other change of a parameter shall be possible without breaking a seal, or the event counter may restart the numbering.

6.2.3.4 Parameters

Legally relevant parameters shall be secured and protected against accidental, unintentional or intentional changes.

Documentation: The documentation shall contain a list of all legally relevant parameters. There shall be no hidden or undocumented legally relevant parameters.

Note: The software identification is a legally relevant parameter.

Examples:

1) (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 protected by a hardware seal.

2) (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 behavior. A hash over all weights in predefined order is used to identify the neural network weights, while a version number is used to identify the overall structure of the neural network as well as the rest of the software. The hash is updated and logged in an audit trail every time the parameters change. The file containing neural weights that matches the hash is stored within the measuring instrument for the time period required by national legislation or stored externally when storage is limited.

3) (I) *Software sealing*. The legally relevant parameters of a measuring instrument can be input and adjusted by a menu item. The software recognizes 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 hardware seal).

4) (I)/(II) *Hardware sealing*. The software of a measuring instrument is constructed such that there is no way to modify the legally relevant parameters except via a switch-protected menu. This switch is set in the inactive position and protected by means of a hardware seal, 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.

5) (II) *Software sealing*. The software of a measuring instrument is constructed such that there is no way to access the legally relevant parameters except by authorized 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 PIN using 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).

Legally relevant parameters that require setting by the user without the need for subsequent verification shall be fitted with an audit trail, see 6.2.3.3.

Documentation: The documentation shall contain a list of those parameters that have to be set by the user.

6.2.3.5 Setting the clock and clock accuracy

Setting the clock, see clause 6.2.2.6 on timestamps, shall be secured and protected against accidental, unintentional or intentional changes.

Guidance: PGs may decide to exempt certain types of measuring instruments from this requirement.

Guidance: PGs may define requirements and test methods for internal clocks in cases where accurate time is required for a legally relevant purpose.

Automatic setting of the time shall only be possible if legal time according to national legislation is used as a time base, in an authenticated manner.

Example:

(I)/(II) The measuring instrument uses NTS protocol in accordance with IETF RFC 8915 to synchronize its clock with an NTS server operated by the national metrology institute. The cryptographic certificate of the NTS server is installed in the instrument during production and treated as a legally relevant parameter, see 6.2.3.4.

Documentation: If a clock is synchronized with legal time, the documentation shall contain a description of the synchronization method and the traceability to legal time.

Note 1: National legislation may establish criteria for an appropriate time reference for ‘legal time’.

Note 2: The term “legal time” refers to the nationally accepted time base for commercial transactions, etc., and is thus subject to national requirements.

6.2.3.6 Measurement data

During processing, measurement data shall be secured and protected against accidental, unintentional or intentional changes.

Note: Protection of the measurement data can be achieved by ensuring that only legally relevant software can process them, and that all requirements for interfaces, see 0, and for configurations specifically, see 6.3, are fulfilled.

6.2.3.7 Interfaces

6.2.3.7.1 Protective interface

It shall not be possible to inadmissibly influence the legally relevant software, parameters or measurement data through protective interfaces.

Each command in the legally relevant software shall be unambiguously assigned to all functions or data changes triggered by it.

Documentation: The documentation shall contain a list of all functions that are triggered through the protective interface.

Only documented functions shall be activated through the protective interface.

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

6.2.3.7.2 User interface

All inputs from the user interface shall be handled by a protective interface.

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 software module is legally relevant.

6.2.3.7.3 Communication interface

All inputs from communication interfaces shall be handled by a protective interface.

6.2.3.7.4 Hardware interface

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 event of inadmissible input, it inhibits measurements.

2) (II) All open interfaces are protected by hardware means or disabled by the operating system.

3) (II) Open interfaces equipped with direct memory access (DMA) are protected by hardware means.

6.3 Requirements specific for configurations

6.3.1 General

Note: The requirements given in 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 measuring instruments that are not software-controlled.

6.3.2 Detection of significant defects

6.3.2.1 General

Guidance: PGs may require functions to detect significant defects, noting that in the case of a software implemented seal, a checking facility is required to check for changes, see 6.3.2.2.

In this case, the manufacturer of the measuring 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.

6.3.2.2 Functional requirements

If software is involved in the detection of significant defects, it shall perform relevant checks at regular intervals.

Guidance: PGs should determine which interval is required for the checks for significant defects.

If software is involved in the detection of significant defects, it shall respond appropriately to any detected defect.

Guidance: PGs should prescribe an appropriate response, e.g., that the instrument or component is deactivated or an alarm/record in an error log is generated if a significant defect is detected.

Note: The checking facility error log is not the same as the audit trail (see 6.2.3.3).

Documentation: The documentation shall contain a list of the significant defects that can be detected by the software, how it will act upon these defects and, if needed for understanding its operation, a description of the detecting algorithm.

Examples:

1) (I) On each start-up, the legally relevant software calculates a checksum of the program code and the legally relevant parameters. The nominal value of this checksum has been calculated in advance and stored in the instrument. If the calculated and stored values do not match, the legally relevant software aborts execution.

   In the case of a non-interruptible cumulative measurement, the checksum is calculated cyclically and controlled by a software timer. If 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 aborts execution.

In the case of a non-interruptible cumulative measurement, the hash value is calculated cyclically and controlled by a software timer. If 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.

6.3.3 Detection of durability errors and/or significant faults

6.3.3.1 General

Note: It is the manufacturer's choice to realize detection of significant faults and durability protection facilities addressed in OIML D 11:2026 [3] (5.1.3 (b) and 5.4) in software or hardware, or to allow hardware facilities to be supported by software.

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.

Guidance: PGs may require functions to detect durability errors and significant faults.

In this case, the manufacturer of the measuring instrument shall be required to design detection functions 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.

6.3.3.2 Functional requirements

If software is involved in durability protection or the detection of significant faults, it shall perform relevant checks at regular intervals.

Guidance: PGs should determine which interval is required for the checks for durability errors and significant faults.

If software is involved in durability protection or the detection of significant faults, it shall respond appropriately to any detected durability error or significant fault.

Guidance: PGs should prescribe an appropriate response, e.g., that the instrument or component is deactivated or an alarm and/or record in an error log is generated if durability is detected as being jeopardized or a significant fault is detected.

Documentation: The documentation shall contain a list of the durability errors and significant faults that can be detected by the software, how it will act upon these errors and faults and, if needed for understanding its operation, a description of the detecting algorithm.

6.3.3.3 Securing and protection

If an error log is used to fulfill requirements from 6.3.3.2, error logs are part of the legally relevant software and shall be secured and protected as such against accidental, unintentional or intentional changes.

It shall not be possible to change the data of an entry in an error log, including when software is updated.

It shall not be possible to delete an entry in an error log, except as an appropriate response to free up storage capacity to allow for the addition of new entries.

Entries shall be recorded automatically.

Any change to the recorded data in the error log, except for deletions to free up storage capacity (as specified above), is a significant software defect and shall be handled accordingly (see).

6.3.4 Dynamic modules of legally relevant software

6.3.4.1 Functional requirements

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 software modules in the measurement process. This may be achieved by the use of a short statement, clearly understood markings, symbols or similar indications. This information, indicating the use of dynamic modules, is regarded as measurement result relevant data.

Documentation: The documentation shall contain a 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 training datasets used, enabling assessment of the algorithm's compliance with the relevant Recommendation.

Guidance: PGs may decide not to implement this requirement in their Recommendation.

6.3.4.2 Securing and protection

The measuring functions shall not be inhibited nor affected by a continuous learning process.

Documentation: The documentation shall contain the description of the prioritization of using all legally relevant software modules, including learning facilities.

Certificate: The certificate shall describe the impact of such dynamic modules on the legally relevant software (modules/parts/algorithms etc.).

It shall not be possible to make any modifications to parameters during a measurement.

Documentation: If dynamic modules of legally relevant software have facilities for continuous learning that allow dynamic parameter changes during use, the documentation shall describe the facilities and the levels of priority assigned to all legally relevant software modules, especially in reference to the measuring functions.

Changes of predefined parameters within dynamic modules of legally relevant software shall be protected, e.g., logging of all parameter changes in an audit trail.

Guidance: PGs should decide if a subsequent verification is required when a legally relevant parameter is changed by the dynamic modules of legally relevant software. To allow for the possibility of parameter adaptations in dynamic modules of legally relevant software without subsequent verification, the source of the parameter change (e.g., the learning facility) is logged in the audit trail, see 6.2.3.3.

6.3.5 Compatibility of operating systems and hardware

6.3.5.1 General

If an operating system is part of the measuring instrument, requirements specified in 6.3.5.2 through 6.3.5.3 shall be met.

Each of the following operating system requirements shall be met by means on application level, operating system level or a combination of both.

Example:

(I)/(II) The protective interface may be implemented within the legally relevant application, the operating system, the physical layer, etc.

Documentation: The documentation shall describe the software identification and the protection and securing means of the operating system as specified in clauses 6.3.5.2 and 6.3.5.3 respectively.

6.3.5.2 Functional requirements

6.3.5.2.1 Software identification

The configuration of the operating system shall be made identifiable as described in 6.2.2.1.

The identifier shall be displayed on command or during operation and, if applicable, transmitted to the verification software by the measuring instrument.

Examples:

1) (I)/(II) On a UNIX-type operating system, the configuration consists of legally relevant

- kernel (/boot/ kernel modules (/lib/modules,
- list of installed packages (/bin/, /sbin/, /usr/bin/, /usr/sbin/, /usr/local/bin/, /etc/init.d/, /etc/systemd/, /usr/lib/systemd/),
- libraries (/lib/ and /usr/lib/),
- accounts and user privileges (/etc/passwd/, /etc/group),
- passwords (/etc/shadow/),
- configuration files (/etc/iptables/, /etc/ufw/, /etc/network/, /etc/ssh/),
- file read/write/execute permissions,
- access settings to interfaces (/usr/lib/udev/, /etc/udev/).

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 (C:\Windows\System32\),
- list of installed packages (C:\Program Files\, C:\Program Files (x86)\),
- libraries (C:\Windows\System32\drivers\),
- accounts and user privileges (C:\Windows\System32\GroupPolicy\),
- passwords,

- configuration files (HKLM\SYSTEM\CurrentControlSet\Services, (HKLM\SYSTEM\CurrentControlSet\Services\SharedAccess\),
- file read/write permissions,
- registry keys,
- access settings to interfaces (HKLM\SYSTEM\CurrentControlSet\Enum\USB).

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

6.3.5.3 Securing and protection

Note: Operating system functions can support the legally relevant task or can influence it. Generally, a general-purpose operating system delivers a complex application programming interface (API) for the legally relevant application,

- to deliver access to specialized hardware (e.g. sensors),
- to start modules in a predefined order and environment,
- to provide managed memory and persistent storage,
- to provide a time base and task scheduling,
- to grant access to users and secure administration tasks.

6.3.5.3.1 Configuration and administrative settings

Legally relevant configuration settings of the operating system shall be protected.

Note: Replacing one legally relevant operating system part with a different one, i.e., with a newer version, is considered a modification of the configuration. This implies that legally relevant operating system parts can only be changed by means of a verified update (see 6.3.9.3), or by means of a traced update (see 6.3.9.4) under the condition that an audit trail is used for protection of the legally relevant configuration settings.

Example:

(I)/(II) All changes to the operating system configuration are logged in an audit trail. Each entry of the audit trail contains a timestamp of the modification as well as the identifier of the new configuration. The software module in charge of maintaining the audit trail and protecting it against modification serves as a trust anchor and is not updated itself, see 6.3.9.4.2.

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, except for temprary files that change during runtime. Folders, accessible via external interfaces, are not writable and executable at the same time.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.

6.3.5.3.2

6.3.5.3.3 Protection during use

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

6.3.5.3.4 Boot process

If a secure boot process is needed to ensure protection of the legally relevant software, this clause shall apply.

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 integrity and authenticity of the legally relevant software, the processing of the chain of trust may be interrupted, but the integrity shall be preserved.

Note: A chain of trust from the protected hardware to the loaded, legally relevant software serves the purpose of ensuring integrity and authenticity of the legally relevant software via mutual authentication of the individual software modules.

The boot configuration shall be secured and protected.

Examples:

1) (I) The boot loader is protected by a device-specific password that is protected by means of a hardware seal inside the housing of the instrument. The sealed housing, together with protection of all open interfaces, ensures that the boot configuration can only be modified after a hardware seal has been broken.

2) (II) A TPM (trusted platform module) verifies the digital signature of the boot loader, the boot loader then verifies the operating system, which in turn verifies and starts the legally relevant application.

Booting via open interfaces shall be disabled.

6.3.5.3.5 Communication with the legally relevant software

Communication with the legally relevant software shall take place via protective interfaces.

Example:

1) (I) A legally relevant software module interprets all commands received by the legally relevant software and discards the inadmissible ones.

2) (I) Firewall settings limit open network ports of the operating system to only allow communication with the legally relevant software through a limited list of protocols.

Note: Regarding interfaces, see 0.

6.3.5.3.6 Suitable environment and constraints for operation

Insufficient resources or an unsuitable environment shall not inadmissibly influence the measurement result. If insufficient resources or an unsuitable environment are detected by the instrument, it shall respond appropriately, see 6.3.2.

Examples:

1) (I)/(II) Technical means provided in the legally relevant software prevent operation if the minimum resources or a suitable configuration are not met.

2) (I)/(II) The minimum number of operating system parts necessary to execute the measurement process is used.

3) (I)/(II) Means are provided to keep the operating environment fixed.

4) (I)/(II) The instrument is designed so that failures due to a lack of resources are treated as significant software defects and acted upon accordingly.

5) (I)/(II) The measurement application on a universal device checks the configuration of the operating system and does not perform any measurements if the operating system does not comply with a predefined (suitable) configuration, such as a specific kernel version.

Documentation: The documentation shall identify a suitable hardware and software environment and minimum resources required.

Certificate: Minimum resources and a suitable software 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.

Guidance: PGs may require the hardware, operating system, or system configuration of a universal device to be fixed or even exclude the usage of an off-the-shelf universal device in the following cases:

- if there is a raised risk level;
- if cryptographic algorithms or keys need to be implemented (see 6.3.6 and 6.3.7).

Note: Regarding inadmissible influence through legally non-relevant software, see 6.3.8.3.3.

6.3.6 Data storage

6.3.6.1 General

Note: Requirements of 6.3.6.2 and 6.3.6.3 regarding data storage apply to software identification, error logs, and, if applicable, to results of diagnostics, results of remote verification and measurement data before they are used for legal purposes.

Guidance: For different applications, PGs may decide if storage of measurement data is required and if additional data need to be stored.

6.3.6.2 Functional requirements

6.3.6.2.1 Completeness of stored data

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

Guidance: PGs should decide which measurement data, e.g., measurement result relevant data necessary to reconstruct the measurement result, shall be stored.

Example:

(I)/(II) A stored dataset of the measurement result includes the following entries:

- measured value including unit;
- timestamp of measurement (see 6.2.2.6);

- place of measurement;
- 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;
- mark showing that the result originates from a dynamic module of legally relevant software, if applicable.

6.3.6.2.2 Automatic storing

Data shall be stored automatically.

A checking facility shall regularly check the availability of the storage and if the storage device is not available or full, this constitutes a significant defect and shall be handled accordingly, see 6.3.2.2.

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

When the measurement 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.

Guidance: PGs should decide which measurement data are relevant for legal purposes.

Measurement data stored in a component to construct the measurement result may be deleted if the next software module or component has checked and stated a proper completion of all expected actions.

6.3.6.2.3 Deletion of the stored measurement result

The measurement result may be deleted if

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

Guidance: PGs should decide how long records that store results of a remote verification shall be kept.

Note: Other general national legislation (e.g., for tax purposes) may contain strict limitations for the deletion of stored measurement data or results.

Guidance: PGs may define alternative conditions for data deletion.

6.3.6.3 Securing and protection

The stored data shall be protected against accidental, unintentional, or intentional changes.

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 hardware seal is broken.

Note: Operating system means can be used to protect sensitive cryptographic information against inadmissible retrieval.

Guidance: PGs may require a raised risk level when considering a freely accessible storage, i.e., storage that is accessible without violating securing and protection means.

Examples:

1) (I) The program of the storing device calculates a CRC32 [9] 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 [9]. 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 the one stored in the dataset. If both values match, the dataset has not been manipulated. Otherwise, the program assumes falsification and discards the dataset.

2) (II) The storing program that is part of the legally relevant software generates a digital 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 by a combination of hardware and software means. The reading program verifies the digital 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 fingerprint of 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 digital 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 digital signature before indicating it.

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

The software that displays, or further processes, the measurement data shall check the authenticity and integrity of the data after having read the them from the storage. If an irregularity is detected, an appropriate response shall be required.

Guidance: PGs may specify appropriate responses to detected irregularities in stored data, e.g., the data shall be discarded or marked as unusable.

Examples:

(I)/(II) A storage device is integrated in a measuring instrument/component fully protected and secured by hardware means. In this case, a standard protocol to enable integrity checks of stored data is used. Authenticity is guaranteed because the housing of the measuring instrument/component is protected by a hardware seal.

2) (I)/(II) In the case of network-attached storage devices or storage in components with limited functionality and protection capabilities, digital signatures are used that enable the retrieving software to check the integrity and authenticity of the records. Means are provided whereby cryptographic keys used by these methods can only be input or read if a seal is broken.

Guidance: PGs may set limitations on storage solutions, e.g., whether or not data shall be stored locally, in different locations or in the cloud.

Example:

1) (I)/(II) For a measuring instrument performing continuous measurements, measurement process data are buffered locally until a measured quantity value associated with a registration interval has been calculated. This value is stored on a cloud server.

6.3.7 Data transmission

6.3.7.1 General

Requirements specified in 6.3.7.2 through 6.3.7.4 regarding data transmission apply to software identification, error logs, results of diagnostics, data transfer during remote verification, measurement data before they are used for legal purposes, etc.

6.3.7.2 Functional requirements

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

Guidance: PGs should decide which measurement data (e.g., measurement result relevant data necessary to reconstruct the measurement result) shall be transmitted.

Example:

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

- measured value including unit;
- timestamp of measurement (see 6.2.2.6);
- place of measurement;
- 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;
- mark showing that the result originates from a dynamic module of legally relevant software, if applicable.

6.3.7.3 Securing and protection

The transmitted data shall be protected by software means to guarantee authenticity and integrity.

Guidance: PGs may require a raised risk level when considering a publicly accessible 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.

Note: 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., digital signatures for transmitted or indicated measurement data) may be used to ensure that produced measurement data are authentic, confirming the software is functioning as intended.

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

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 response shall be required.

Guidance: PGs should decide what response is required, e.g., the measurement data shall be discarded or marked as unusable.

Examples:

1) (I)/(II) A component is directly connected and sealed to another component. In this case, a standard protocol to enable integrity checks of transmitted data is used.. Authenticity is guaranteed because the component is protected by a hardware seal to prevent exchange.

2) (I)/(II) In the case of network-attached components, the legally relevant software of the sending device calculates a CRC32 [9] of the dataset, which is appended to the dataset. A secret initial value is used for the calculation of the CRC32 instead of the value given in the standard [9]. 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 the one stored in the dataset. If both values match, the dataset is not falsified. Otherwise, the program assumes falsification and discards the dataset.

3) (II) The legally relevant software of the sending device generates a digital signature for the transmitted dataset. This digital signature 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 by hardware and software means and exports the public key. The legally relevant software of the receiving device verifies the digital 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.

4) (I)/(II) In the case of web-based components and components with limited functionality and protection capabilities, digital signatures are used that enable the retrieving software to check the integrity and authenticity of the records. Means are provided whereby cryptographic keys used by these methods can only be input or read if a seal is broken.

6.3.7.4 Transmission delay or interruption

The measurement shall not be inadmissibly influenced by a transmission delay, or by the interruption or unavailability of network services. If a transmission delay or the interruption or unavailability of network services occurs, an appropriate response shall be required.

Guidance: PGs should decide what response is required, e.g., disabling the measurement function, stopping the current measurement process, discarding the measurement result or marking it as unusable.

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 measurement data may be acceptable, provided this is detected and the user is informed that measurement data have been lost.

Examples:

1) (I)/(II) The sending measuring instrument or component waits until the receiver has sent a confirmation that the dataset has been received correctly. The sending instrument or component keeps the dataset in a buffer until this confirmation has been received. The buffer has a capacity for more than one dataset, organized as a FIFO (First-in-first-out) queue.

2) (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 FIFO order. If the local buffer reaches its limit, further measurements are disabled.

6.3.8 Specification and separation of legally relevant components and software modules

6.3.8.1 General

These requirements apply if a measuring instrument contains separate components or software modules.

Guidance: PGs may specify the software modules, components or parts of the software modules or components that are legally relevant.

6.3.8.2 Specification and separation of components

6.3.8.2.1 General

Documentation: Components of a measuring instrument that perform legally relevant functions shall be identified, clearly defined and documented. They form the legally relevant hardware of the measuring instrument.

Note 1: Regarding separation of software modules, see 6.3.8.3.

Note 2: 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.

Examples:

1) (I)/(II) An electricity meter with a local display is equipped with a protective optical interface for connecting a 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 measuring instrument. Other (legally non-relevant) devices can be connected to the protective interface that complies with 6.2.3.7.1. Securing of the data transmission itself (see 6.3.7) 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 via 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 legally non-relevant. The legally relevant components fulfil requirement 0 and – because of the transmission via the network – also the requirements contained in 6.3.7.

6.3.8.2.2 Shared components

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

Guidance: PGs should decide if it is required to identify components on a print-out. This may not be necessary in cases where the product bears a label or the measurement is repeatable.

Note 2: If a measurement is repeatable, there is no need to identify the exact components that produced a measurement result in the printout. If the measurement cannot be repeated, such identification on a printout enables inspectors etc. to check for the source of an error.

6.3.8.2.3 Securing and protection

Legally relevant components shall be protected against exchange.

Guidance: PGs may decide to exempt some components from this protection requirement against exchange, e.g., in the case of simple recipient printers.

Note: Regarding interfaces, see 0.

Examples:

1) (I)/(II) The software of an electricity meter with a protective optical interface for connecting other devices is able to receive commands for selecting the measurement results required. It sends the measurement result (including additional measurement result relevant data – e.g., timestamp, 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) A measuring instrument consists of two components communicating with each other. Both components are placed together in the same housing and the communication cable is protected by a hardware seal within the housing to prevent tampering.

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, see 6.2.3.4.

Note: In general, pairing parameter means any parameter that is necessary to connect and run the separated components that form the measuring instrument, such as network or internet (IP) address, Bluetooth pairing key, and encryption key. Depending on the individual design of the measuring instrument, this includes parameters that are used as part of a software seal to prevent exchanging or spoofing components.

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 set manually. Communication is encrypted in both directions using AES-128 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 digital 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 digital signature of the sender cannot be verified, the receiver displays an error message and prevents further measurements.

Legally relevant components shall check the authenticity, integrity and/or availability of other legally relevant software-controlled components. When the authenticity and/or integrity check fails, or the other component is not available, the checking component shall respond appropriately.

Guidance: PGs should decide which action shall be taken if the authenticity and/or integrity check fails.

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

Example:

(I/II) If the indication of a result is mandatory, a display is connected and available to the measuring instrument.

Guidance: PGs may decide to exempt some components from this checking requirement for authenticity, integirty and availability, e.g., in the case of simple recipient printers it could be that only availability needs to be checked.

Legally non-relevant components or devices shall be prevented from calculating/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 are 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 1) in 6.3.8.3.3, detailing encryption of measurement data from the measurement sensor.

If legally relevant components have limited functionality and limited securing/protection capabilities (e.g., if a legally relevant operating system on a component cannot be configured in accordance with 0), they shall have limited access to the measurement result relevant data, i.e., they shall only indicate the measurement data without modifying them.

- 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 receiving component shall be capable of checking the authenticity and integrity of the measurement data.

If increased protection against fraud is necessary, a component shall exist with increased securing means that can display or print the measurement results in the event of a dispute.

Example:

(I) The measurement is started remotely by a mobile app, which runs on a dedicated device belonging to the owner of the measuring instrument. The instrument itself is fully secured and protected (both by hardware and software means) 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 mobile device such as a smartphone for indication as clear text accompanied by a two-dimensional bar code that contains measurement result and digital signature. In case of doubt, the correct indication of the result can be checked by all parties by validating the digital signature contained in the two-dimensional bar code, see also 6.2.2.5. The signed measurement result can be uploaded to a secured webserver protected by software means that checks the digital signature and then indicates the result.

6.3.8.3 Specification and separation of software modules

6.3.8.3.1 General

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.

Legally relevant software modules shall be made identifiable as described in 6.2.2.1.

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

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

- For separation of components, see 6.3.8.2.
- For communication between multiple legally relevant components, see 6.3.7.

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 realizing 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.

Note: If one or more dynamic modules of legally relevant software are used in combination with software separation, 6.3.4.2 needs to be observed to ensure that any parameter changes in these software modules are logged in the audit trail.

6.3.8.3.2 Mixed software identification

If the manufacturer chooses a mixed identifier for legally relevant and legally non-relevant software, the legally relevant software identifier(s) shall be clearly distinguishable from the legally non-relevant part. In this case, applicable requirements are given in 6.2.2.1.

6.3.8.3.3 Securing and protection

Legally non-relevant software modules shall be prevented from calculating/presenting/spoofing the measurement result.

Note: This does not preclude legally relevant software modules from presenting measurement process data.

All legally relevant software modules shall communicate with other software modules and/or components through a protective interface, see 6.2.3.7.1.

Examples:

1) (I) In a measuring instrument that realizes software separation, 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 can read and decrypt measurement data, and display measurement results. Only after indication of the measurement results does the library allow other (legally non-relevant) software modules to read the result.

2) (I) In a measuring instrument, the protective 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 are fixed at compile time.

3) (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 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, i.e., a secret password written on a label within the sealed housing of the instrument. Therefore, changes to the setup of the virtual machines cannot be made without breaking a seal.

4) (I)/(II) Measurement data is not made available to legally non-relevant software modules prior to primary indication.

The legally relevant process shall not be inadmissibly interrupted by legally non-relevant software.

Example:

(I)/(II) The operating system grants priority to the legally relevant software over the legally non-relevant software when using resources.

The measurement process (realized by the legally relevant software) shall not be delayed or blocked by other processes.

Examples:

1) (I) Priority levels are assigned to the legally relevant functions, which are higher than for legally non-relevant 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 measuring instrument is able to communicate via an interface with other devices in parallel, served by another interrupt routine (legally non-relevant communication). The interrupt routine for processing the raw values has a higher priority 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 running the legally relevant software always has sufficient system resources available for the legally relevant processes.

Documentation: The documentation shall contain a description of the levels of priority assigned to all legally relevant software modules and all legally non-relevant software modules. Furthermore, it shall contain a list of all legally relevant software modules and the protective interface. All legally relevant functions and data domains of the software shall be described to enable a type evaluation authority to decide if software separation has been implemented correctly.

6.3.9 Maintenance and reconfiguration

6.3.9.1 General

The following options, verified update 6.3.9.3 and traced update 6.3.9.4, are alternatives.

If device-specific parameters (for example adjustment parameters) are modified during an update, only a verified update is permitted.

Guidance: PGs should decide whether a verified or a traced update is allowed.

Note 1: This issue concerns the verification of a measuring instrument in the field. Refer to clause 8 for additional constraints.

Note 2: Software that does not realize legally relevant functions of the measuring instrument does not require verification after being updated.

Documentation: The documentation shall contain a list of all components that form the legally relevant hardware.

Certificate: The components that form the complete legally relevant hardware shall be stated in the certificate.

6.3.9.2 Securing and protection

An update shall not inadmissibly influence the measurement process.

6.3.9.3 Verified update

6.3.9.3.1 Functional requirements

Note 1: The software to be updated may be loaded locally, i.e., directly on the measuring instrument, or remotely via a network.

Note 2: Loading and installation may be two different steps (as shown in Figure 1) or combined into one, depending on the needs of the technical solution.

6.3.9.3.2 Securing and protection

Access to the verified update shall be protected by hardware or software seals that must be broken for the update to take effect.

Note: After the update of the legally relevant software of a measuring instrument (exchange with another approved software version or re-installation), the securing and protection means should be renewed and the measuring instrument should be verified, see 8.2 for a list of verification methods.

Guidance: PGs may also specify other procedures following a verified update.

*Certificate:*The means of how the protection means are renewed or reactivated, if different from the normal securing or protection activation method, shall be stated in the certificate.

6.3.9.4 Traced update

6.3.9.4.1 General

Guidance: 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.

Certificate: The certificate shall state if the instrument is capable of performing a traced update.

The software shall be implemented in the instrument in accordance with the requirements for traced updates (6.3.9.4.2 to 0).

6.3.9.4.2 Functional requirement - automatic update procedure

After initiation of the update procedure, a traced update of software shall run automatically.

Note 1: As shown in Figure 1, the procedure of a traced update comprises several steps: loading, integrity checking, authentication (checking the origin), installation, logging and activation.

Note 2: The software to be updated may be loaded locally, i.e., directly on the measuring instrument, or remotely via a network.

Note 3: National legislation may require a feature for the user or owner of the device to express their consent prior to an update.

Guidance: PGs should decide if it is necessary for the user or owner to express their consent prior to an update, e.g., by means of a push button.

If a feature is required for the user or owner to express their consent prior to an update, 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.

The software update shall be recorded in an audit trail.

A traced update shall not affect existing parameters.

6.3.9.4.3 Functional requirement - data continuity

A traced update shall not influence the legally relevant parameters.

Note: The software identification will change during an update, even if it is treated as a legally relevant parameter, see 6.2.3.4.

If some of the securing or protection means of the measuring instrument are turned off to enable updating, they shall be turned on again automatically immediately after the update, regardless of the result of the update process.

During or as a result of a traced update, any existing protection means, 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.

6.3.9.4.4 Functional requirement - software integrity and authenticity

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 (usually the manufacturer of the measuring instrument) generates a digital signature of the revised software or software 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 digital signature is checked using the public key when loading the revised software into the measuring instrument. If the digital signature of the loaded software is correct, 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.

Technical means shall be employed to ensure the integrity of the loaded software, i.e., that it has not been inadmissibly changed before loading.

Example:

(I)/(II) A checksum or hash code over the loaded software is verified during the loading procedure.

If the loaded software fails the integrity test or the authenticity test, the measuring 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. Only two things shall be possible: Either the user can manually restart the download procedure or the instrument displays an error message.

6.3.9.4.5 Securing and protection

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 previously installed version;
- timestamp of the event;
- identification of the uploading party, i.e., the source of the update, e.g., operator, service engineer or manufacturer, if available.

Note 1: An entry is generated for each update attempt regardless of success.

The storage device that supports the traced update shall have sufficient capacity to ensure the traceability of traced updates of the legally relevant software between at least two successive verifications or inspections of a measuring instrument in the field.

If the audit trail has no more capacity, an appropriate response is required.

Note 2: This requirement enables inspection authorities that 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).

Guidance: PGs should specify a sufficient capacity for the audit trail and the required response, i.e., either the oldest entry may be deleted or the update procedure should not start at all.

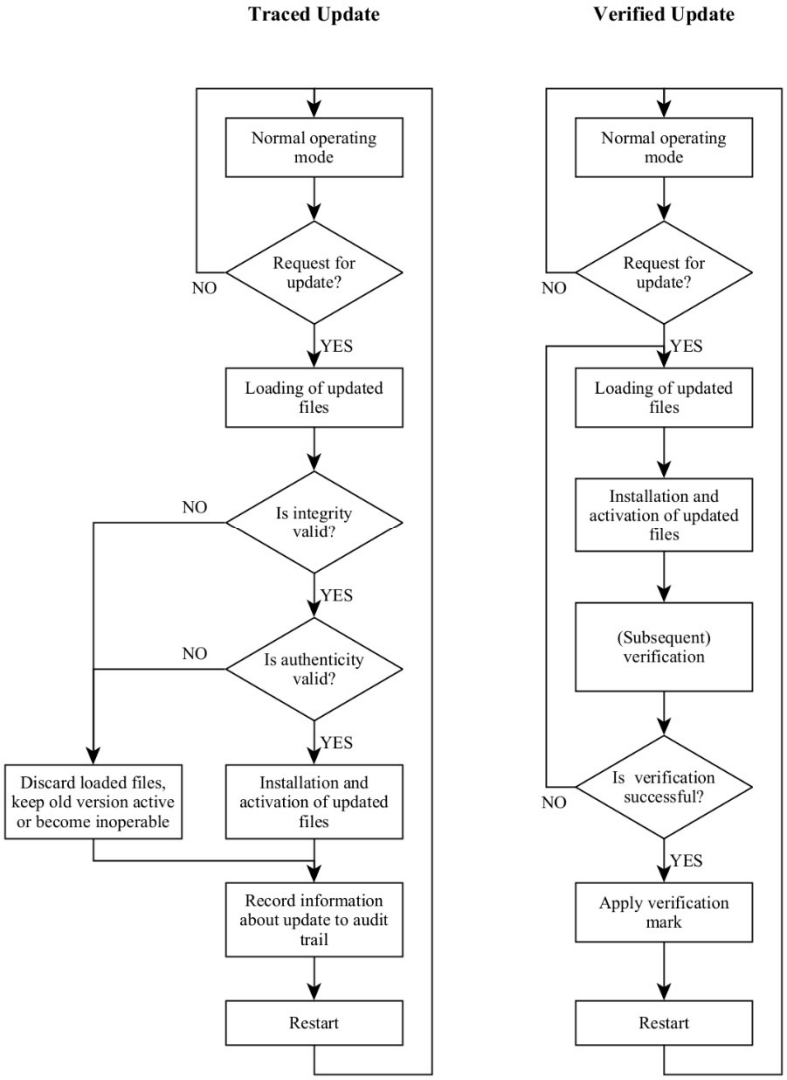

Figure 1 - Software update procedure

Note 3: 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 4: 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 5: In this figure, only the 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, symbolized by the NO-branch.

6.3.10 Remote verification capability

6.3.10.1 General

If the measuring instrument facilitates remote verification, the requirements in 6.3.10.2 and 6.3.10.3 shall be met.

Documentation: The documentation shall contain a description of the remote verification procedure for accessing/reading of remote verification data and for executing remote verification procedures.

Note 1: The description may need to be made available to the relevant authorities depending on national legislation.

Note 2: National legislation may allow or disallow remote verification.

6.3.10.2 Functional requirements

6.3.10.2.1 General

For the purpose of remote verification, the measuring instrument shall

- use timestamps (6.2.2.6),
- provide evidence of an intervention (6.2.3),
- use audit trails (6.2.3.3), store logging data,
- have a facility for detection of significant defects (6.3.2), and
- make these data and functionalities available via a communication interface.

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

It shall be possible to establish and ensure the integrity of the instrument to be verified, including its software module that sends data.

Example:

(I)/(II) The instrument engages with a verifier in a software remote attestation protocol. The instrument receives a random challenge from the verifier, calculates a checksum of the executable code (including the software module for data transmission) concatenated with the challenge, and presents the result. The verifier checks the outcome of the computation against a reference value.

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

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.

Example:

(1)(II) A measuring 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, communication is established.

Remote verification functionality shall be secured.

6.3.10.2.2 Direct extraction of test items

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

Relevant test items shall be available depending on the specific requirement to be tested and the instrument type.

Guidance: PGs should define a list of relevant test items for verification purposes, e.g., approved type number, serial number, legally relevant settings and parameters, verification information and status, software identification, software integrity, audit trails, error logs etc.

Note: See subclauses of 8.3.3.2 for examples of test items for a specific remote verification procedure.

6.3.10.2.3 Result of the remote verification

The result of the remote verification shall contain, at least, a unique ID (at least identifying the verification authority) and the date of the verification.

Guidance: PGs should decide which additional data shall be stored.

Note: The recognition of a verification mark and the data it contains may be subject to national legislation.

6.3.10.3 Securing and protection

Interfaces for remote verification shall be protected.

Note: General guidance on interface protection may be found in 0.

The data transmission to the remote verification software shall comply with 6.3.7.

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

An ongoing measurement shall not be influenced by remote verification.

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

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

The access to the verification procedure, specific test items or commands shall be restricted if these influence compliance with other requirements, such as:

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

Guidance: PGs should decide if access to the verification procedure shall always be restricted.

Provisions shall be made to securely store the result of the remote verification in the measuring instrument. These data shall be protected and secured.

Stored results of the verification in the instrument shall comply with 6.3.6.

Securing shall ensure that only the remote verification software has write permissions.

Documentation: The documentation shall contain a description of access rights to the instrument for remote verification. These rights shall be made available to the relevant authorities depending on national legislation.

After the minimum storage period for results of a remote verification has elapsed and if the storage device has no more capacity, the oldest entry of records may be deleted.

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 in full 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: For dynamic modules of legally relevant software (e.g., evolving machine learning models), the documentation shall describe clear ways of verification and evaluation of said dynamic modules.

Guidance: Regarding 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 following list is a summary of all documentation requirements from clauses 4.4, 6.2 and 6.3. The documentation (for each measuring instrument or component) shall at least include:

- the operating manual;
- description of all legally relevant functions and how to access them, see 6.2.2.2;
- list of all legally relevant parameters, see 6.2.3.4, and a description of protection and securing means;
- list of all parameters that have to be set by the user, see 6.2.3.4;
- if a clock is synchronized with legal time, the synchronization method and traceability to legal time, see 6.2.3.5;
- description of the legally relevant software and how the requirements are met:
    - list of all legally relevant software modules, see 6.3.8.3.3;
    - description of the protective interface, see 6.3.8.3.3, and all functions that are triggered through the protective interface, see 6.2.3.7.1;
    - 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;
- description of suitable system configuration and minimum resources required, see 6.3.5.3.6;
- description of the software identification and the protection and securing means of the operating system (e.g., password), see6.3.5.1;

- description of the suitable hardware and software environment, see 6.3.5.3.6;
- 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 and clearly defined, see 6.3.8.2.1;
- description of the accuracy of the algorithms, see 6.2.2.2 (e.g., filtering of A/D conversion results, price calculation, rounding algorithms, etc.);
- description of the user interface, menus and dialogs;
- software identification and instructions for obtaining it from an instrument in use, see 6.2.2.1;
- list of all commands of each hardware interface of the measuring instrument or component, see 6.2.3.1;
- list of all durability errors and significant faults that are detected by the software, how it will act upon these errors and faults and, if needed for understanding its operation, a description of the detecting algorithm, see 6.3.3.2;
- the required metadata for legally relevant measurement data, see 4.4;
- description of datasets stored or transmitted;
- if detection of significant defects is realized in the software, a list of the significant defects that will be detected by the software, how it will act upon these defects and, if needed for understanding its operation, a description of the detecting algorithm, see 6.3.2.2;
- if an audit trail is realized in the software, a description of how to access the audit trail;
- list of all components of a measuring instrument that perform legally relevant functions, see 6.3.8.2.1;
- complete list of components that form the complete legally relevant hardware, see 6.3.9.1;
- if software separation is implemented:
    - description of all legally relevant software functions and data domains, see 6.3.8.3.3;
    - list of all legally relevant software modules and the protective interface, see 6.3.8.3.3;
    - description of the levels of priority assigned to all legally relevant software modules and all legally non-relevant software modules, see 6.3.8.3.3;
- 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, see 6.3.10.1;
    - 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, see 6.3.10.3;
- if dynamic modules of legally relevant software are present:
    - when dynamic modules of legally relevant software have facilities for continuous learning that allow dynamic parameter changes during use, a description of these facilities and the levels of priority assigned to all legally relevant software modules, especially in reference to the measuring functions, see 6.3.4.2;

- a description of the prioritization of using all legally relevant parts, including dynamic modules of legally relevant software, see 6.3.4.2;
- a description of the means to validate the conformity of devices in use even in the presence of dynamic parameter changes, see 6.2.1;
- detailed description of the dynamic module's algorithm design as well as a description of the training process and the used training datasets, see 6.3.4.1.

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 25040:2024 [10]]. 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 optimizes 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 approved 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 for more information.

The methods specified for software evaluation are described in clause 7.3. Combinations of these methods to form a complete software evaluation procedure that is adapted to all requirements defined in clause 6 are specified in clause 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 their 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:

- the software identification and the means of identification (e.g., software version, hash value, checksum, CRC), see 6.2.1 and 6.2.2.1.
- Instruction on how the software identification, see 6.2.2.1, relevant parameter settings, and evidence of an intervention may be displayed or printed, and specify how it can be obtained by the remote verification procedure, see 6.2.2.7;
- securing means as well as means to provide evidence of an intervention and the method to check them (e.g., hardware seals, event counter(s), audit trail(s)), see 6.2.3;

- software modules under legal control, including whether or not the instrument is equipped with a remote verification procedure (see 8.3.2) or a traced update procedure, see 6.3.9.4;
- specification whether the measuring instrument is equipped with dynamic modules and their impact on the legally relevant software (modules/parts/algorithms etc.), see 6.3.4.2.
- Minimum resources and a suitable software 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, see 6.3.5.3.5;

- if applicable:
  - means of how the protection means are renewed or reactivated, see 6.3.9.3;
  - test items with their unique identification used for the remote verification procedure, see 6.3.10.2.2.

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 from case to case in a software evaluation procedure.

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 documentation and specification and evaluation of the design (7.3.2.1)	Always	Documentation	-
VFTM	Verification by functional testing of the metrological functions (0)	Correctness of the algorithms, uncertainty, compensating and correcting algorithms, rules for price calculation	Documentation, specimen	-

VFTSw	Verification by functional testing of the software functions (7.3.2.3)	Correct functioning of communication, indication, evidence of an intervention, protection against operating errors, protection of parameters, detection of significant defects	Documentation, specimen	-
DFA	Metrological dataflow analysis (7.3.2.4)	Software separation, evaluation of the impact of commands on the instrument's functions	Source code, tools for analyzing source code	Knowledge of programming languages
CIWT	Code inspection and walk through (7.3.2.5)	All purposes	Source code, tools for analyzing 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
6.2 General requirements				
6.2.1	Conformity of manufactured devices to the approved type	AD	AD	
6.2.2 Functional requirements				
6.2.2.1	Software identification	AD + VFTSw	AD + VFTSw + CIWT	Select “B” if high conformity is required
6.2.2.2	Correctness of algorithms and functions	AD + VFTM	AD + VFTM + CIWT/SMT	
6.2.2.3	Prevention of misuse	AD + VFTSw	AD + VFTSw + DFA/CIWT/SMT	Select “B” in case of high risk of fraud
6.2.2.4	Indications	AD + VFTM/ VFTSw	AD + VFTM/VFTSw + DFA/CIWT	
6.2.2.5	Shared indications	AD + VFTM/ VFTSw	AD + VFTM/VFTSw + DFA/CIWT	
6.2.2.6	Timestamps	AD + VFTSw	AD + VFTSw + SMT	
6.2.2.7	Information for verification	AD + VFTSw	AD + VFTSw + CIWT + SMT	Select “B” if high reliability is required
6.2.2.8	Contents of the audit trail	AD + VFTSw	AD + VFTSw + CIWT + SMT	Select “B” if high reliability is required
6.2.3 Securing and protection				
6.2.3.1	General	AD + VFTSw	AD + VFTSw/CIWT+SMT	Select “B” if high conformity is required
0	Software	AD + VFTSw	AD + VFTSw/CIWT+SMT	Select “B” if high conformity is required
6.2.3.3 Means to provide evidence of an intervention				
Error! Reference source not found.	Functional requirements	AD + VFTSw	AD + VFTSw/CIWT	
Error! Reference source not found.	Securing and protection	AD + VFTSw	AD + VFTSw/CIWT+SMT	
6.2.3.4	Parameters	AD + VFTSw	AD + VFTSw/CIWT	

Requirement		Examination level A (normal examination level)	Examination level B (extended examination level)	Comment
6.2.3.5	Setting the clock and time accuracy	AD + VFTSw	AD + VFTSw/CIWT	
6.2.3.6	Measurement data	AD + VFTSw	AD + VFTSw/CIWT	
0 Interfaces				
6.2.3.7.1	Protective interface	AD + VFTM	AD + VFTM/VFTSw	
6.2.3.7.2	User interface	AD + VFTM	AD + VFTM/VFTSw	
6.2.3.7.3	Communication interface	AD + VFTM	AD + VFTM/VFTSw	
6.2.3.7.4	Hardware interface	AD + VFTSw	AD + VFTSw + SMT	
6.3 **Requirements specific for** configurations				
6.3.2 Detection of significant defects				
6.3.2.1	General	AD + VFTSw	AD + VFTSw + CIWT + SMT	Select “B” if high reliability is required
6.3.2.2	Functional requirements	AD + VFTSw	AD + VFTSw + CIWT + SMT	Select “B” if high reliability is required
6.3.3 Detection of durability errors and/or significant faults				
6.3.3.1	General	AD + VFTSw	AD + VFTSw + CIWT + SMT	Select “B” if high reliability is required
6.3.3.2	Functional requirements	AD + VFTSw	AD + VFTSw + CIWT + SMT	Select “B” if high reliability is required
6.3.3.3	Securing and protection	AD + VFTSw	AD + VFTSw + CIWT + SMT	Select “B” if high reliability is required
6.3.4 Dynamic modules of legally relevant software				
6.3.4.1	Functional requirements	AD + VFTSw	AD + VFTSw + CIWT/SMT	
6.3.4.2	Securing and protection	AD + VFTSw	AD + VFTSw/CIWT+SMT	
0 **Compatibility of operating systems and** hardware				
6.3.5.2 Functional requirements				
6.3.5.2.1	Software identification	AD + VFTSw	AD + VFTSw + CIWT	Select “B” if high conformity is required
6.3.5.3 Securing and protection				

Requirement		Examination level A (normal examination level)	Examination level B (extended examination level)	Comment
6.3.5.3.1	Configuration and administrative settings	AD + VFTSw	AD + VFTSw/CIWT+SMT	
6.3.5.3.2	Protection during use	AD + VFTSw	AD + VFTM/ VFTSw + DFA+SMT	
6.3.5.3.4	Boot process	AD + VFTSw	AD + VFTSw + SMT	
6.3.5.3.5	Communication with the legally relevant software	AD + VFTSw	AD + VFTM/ VFTSw + DFA+SMT	
6.3.5.3.6	Suitable environment and constraints for operation	AD + VFTSw	AD + VFTSw + SMT	
6.3.6 Data storage				
6.3.6.2.1	Completeness of stored data	AD + VFTSw	AD + VFTSw + CIWT/SMT	Select “B” in case of high risk of fraud
6.3.6.2.2	Automatic storing	AD + VFTSw	AD + VFTSw + SMT	
6.3.6.2.3	Deletion of the stored measurement result	AD + VFTSw	AD + VFTSw + SMT	
6.3.6.3	Securing and protection	AD + VFTSw	AD + VFTSw + CIWT+SMT	
6.3.7 Data transmission				
6.3.7.2	Functional requirements	AD + VFTSw	AD + VFTSw + CIWT/SMT	Select “B” if transmission of measurement data in open system is foreseen
6.3.7.3	Securing and protection	AD + VFTSw	AD + VFTSw/CIWT+SMT	
6.3.7.4	Transmission delay or interruption	AD + VFTSw	AD + VFTSw + SMT	Select “B” in case of high risk of fraud, e.g., transmission in open systems
6.3.8.2 Specification and separation of components				
6.3.8.2.2	Shared components	AD	AD + DFA/CIWT	
6.3.8.2.3	Securing and protection	AD + VFTSw	AD + VFTSw/CIWT+SMT	
6.3.8.3 Specification and separation of software modules				
6.3.8.3.1	General	AD	AD + DFA/CIWT	
6.3.8.3.2	Mixed software identification	AD	AD + DFA/CIWT	
6.3.8.3.3	Securing and protection	AD + VFTSw	AD + VFTSw/CIWT+SMT	

Requirement		Examination level A (normal examination level)	Examination level B (extended examination level)	Comment
6.3.9 Maintenance and reconfiguration				
6.3.9.2	Securing and protection	AD + VFTSw	AD + VFTSw/CIWT+SMT	
6.3.9.3 Verified update				
6.3.9.3.1	Functional requirements	AD	AD	
6.3.9.3.2	Securing and protection	AD + VFTSw	AD + VFTSw/CIWT+SMT	
6.3.9.4 Traced update				
6.3.9.4.1	General	AD + VFTSw	AD + VFTSw + CIWT/SMT	
6.3.9.4.2	Functional requirements – automatic update procedure	AD + VFTSw	AD + VFTSw + CIWT/SMT	Select “B” in case of high risk of fraud
0	Functional requirements – data continuity	AD + VFTSw	AD + VFTSw + CIWT/SMT	Select “B” in case of high risk of fraud
6.3.9.4.4	Functional requirements – software integrity and authenticity	AD + VFTSw	AD + VFTSw + CIWT/SMT	Select “B” in case of high risk of fraud
0	Securing and protection	AD + VFTSw	AD + VFTSw/CIWT+SMT	
6.3.10 Remote verification capability				
6.3.10.1	General	AD	AD	
6.3.10.2 Functional requirements				
6.3.10.2.1	General	AD + VFTSw	AD + VFTSw + CIWT/SMT	
6.3.10.2.2	Direct extraction of test items	AD + VFTSw	AD + VFTSw + CIWT/SMT	
6.3.10.2.3	Result of the remote verification	AD + VFTSw	AD + VFTSw + CIWT/SMT	
6.3.10.3	Securing and protection	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 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.
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.

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 an 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, "verification by functional testing of the metrological functions (VFTM)" (see 0) is a complementary procedure.

Reference:

IEC 61508-5:2010 [11].

7.3.2.2 Verification by functional testing of the metrological functions (VFTM)

Application:

Verifying correctness of algorithms for calculating the measurement result from measurement data, for linearization 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 purpose 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 measuring instrument. Although it does not aim primarily at verifying the software, the test result can be interpreted as a verification of some software modules, 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:

This 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 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 purpose of answering questions.

Description:

Required features described in the operating manual, measuring instrument documentation or software documentation are checked in practice. 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 attempting 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 undocumented commands just by random trial and error. 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 7.2, Sections 4.2 and 5.2 [12].

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 for. This subroutine will store them in a variable after possibly having done some processing. From this variable, the processed value is read by another subroutine and so forth until the completed measurement result is output to the display. All variables that are used for buffering 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 6.3.8.3.3) can be detected.

Result:

It can be verified whether software separation according to 6.3.8.3 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 realized 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 61131-3:2025 [13].

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, etc. 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 they have identified as complex, error-prone, insufficiently documented, etc., and inspect the respective part of the source code by analyzing 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 is or is not compatible with the software documentation and in compliance with the requirements.

Complementary procedures:

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

Reference:

IEC 61508-5:2010 [11].

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. Cooperation with the programmer of the software 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 software module under test and provides it with all necessary input data. The test program receives actual output data from the software module under test and compares them with the nominal values.

Result:

Output of the software module under test is correct or not.

Complementary procedures:

This is an enhanced method, in addition to 0 or 7.3.2.5.

Reference:

IEC 61508-5:2010 [11].

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.

Guidance: PGs should justify the selection of level B 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 approved type, and
- risk of wrong measurement result due to operating errors.

See clause 5 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 providing 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 approved 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 approved software 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 of the operating system, 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.

Guidance: PGs should 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) dataset 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. This does not affect the requirement that instrument software shall be verifiable.

8.2 Verification methods, test items

The following methods comprise the verification steps which are needed to check the requirements of 6.2 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, see 6.2.3.3;
- directly: Check the software identifiers as required in the certificate, see 6.2.2.1.

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(s) 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(s) for traced updates (see 0).

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 a measuring 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).

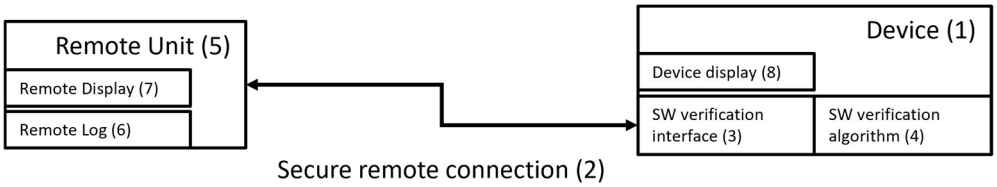

Figure 2 - Remote verification procedure

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.

Note: National legislation may allow or disallow remote verification depending on the instrument.

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

Remote verification shall cover the communication between legally relevant software modules, see 6.3.7. The communication connection between legally relevant software of the measuring instrument and software for verification purposes on the remote unit (see Figure 2) shall be available.

Note: 6.3.10.3 requires that this connection shall comply with 6.3.7, transmission via communication lines.

The integrity and authenticity of the measuring instrument shall always be checked, see 6.3.10.2.1.

Guidance: PGs should define a list of relevant data for verification purposes depending on the instrument type (approved type number, serial number, legally relevant settings and parameters, verification information and status, software identification, software integrity, audit trails, error logs etc.).

Certificate: The certificate shall state that remote verification is foreseen for this instrument and list test items with their unique identification used for the remote verification procedure, see 7.2.2.

Note 1: The remote verification procedure can only be executed when the device to be remotely verified is available and ready.

Note 2: The device needs to be able to execute the verification procedures.

Note 3: This Document only imposes requirements on the measuring instrument's software. Verification software running on the remote unit is covered by national legislation.

The following clause, 8.3.3, describes examples of specific remote verification procedures and lists the test items necessary for those remote verification procedures.

Guidance: PGs should select the appropriate remote verification procedures depending on the type of instrument. Instrument-specific verification procedures (see 8.3.3.3) shall be detailed in the relevant Recommendation.

8.3.3 Examples of specific remote verification procedures

8.3.3.1 Extraction of data from audit trails or other logging mechanisms

The purpose of this remote verification procedure is to check a measuring instrument's operational history by retrieving the logging mechanisms.

Applicable test items for this remote verification procedure are audit trails, event counters etc.

The value of these test items is compared with a reference value.

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 identification, software integrity, audit trails, error 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.

8.3.3.2 Direct extraction of test items

8.3.3.2.1 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 identification) are then compared with a known reference. Relevant test items identified by the PGs shall be available, see 6.3.10.2.2.

Applicable test items for this remote verification procedure are software integrity, correctness of parameters, and 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: It is the manufacturer's obligation to provide information about external SW for performing tests, see also 8.3.1.

8.3.3.2.2 Precondition for direct extraction of test items

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 environment, see 8.3.3.1.

8.3.3.2.3 Software integrity

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

The applicable test item for this remote verification procedure is the integrity measure (checksum, hash).

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

8.3.3.2.4 Check of parameters

The purpose of this remote verification procedure is to check whether the parameters have not been changed (there is no evidence of an intervention) and, if reference parameter values are available, whether they are correct.

The applicable test item for this remote verification procedure is the value of the parameter and the integrity measure of the parameters, i.e., audit trail or event counter.

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

8.3.3.2.5 Software identification

The purpose of this remote verification procedure is to check the software identification.

The applicable test item for this remote verification procedure is the value of the software identification.

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

8.3.3.3 Instrument-specific remote verification procedures

8.3.3.3.1 General

The following subclauses, 8.3.3.3.2 to 8.3.3.3.5, each give an example of a specific realization of this remote verification procedure for specific types of measuring instruments. These procedures shall be secured.

Note 1: It is the manufacturer's obligation to describe the test procedure, the result of which shall be made available to the relevant authorities depending on national legislation, see 6.3.10.

Note 2: It is the manufacturer's obligation to describe the simulation procedure, the result of which shall be made available to the relevant authorities depending on national legislation, see 6.3.10.

8.3.3.3.2 Weighing instrument

Initiate an internal weighing procedure using a built-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.

8.3.3.3.3 Flow meter

Initiate procedure using a built-in diagnostics facility to establish whether the current performance of a flow meter has degraded since the last calibration and 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.

8.3.3.3.4 Digital data processing unit

Simulating a digital sensor and sending measurement process data to the digital data processing unit and retrieving 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.

8.3.3.3.5 Point-to-point speed meter

A test object passes in front of a speed meter, traveling through a corridor of known length. At the beginning of the corridor, a start signal is sent to the remote verification unit. At the end of the corridor, a stop signal is sent to the remote verification unit. The measurement result is retrieved from the point-to-point speed meter and compared with the reference value calculated by the remote unit to evaluate the accuracy of the measurement algorithms of the point-to-point speed meter.

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

Annex A

Bibliography
(Informative)

At the time of publication, the editions indicated were valid. All referenced 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 referenced 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

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 V 1:2022 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.
[3]	OIML D 11:2026 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).
[4]	ISO/IEC 9594-8:2020 Information technology -- Open Systems Interconnection -- Part 8: The Directory Public-key and attribute certificate frameworks	ISO/IEC 9594-8:2020 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.
[5]	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.

Ref.	Standards and reference documents	Description
[6]	ISO/IEC 27005:2022 Information security, cybersecurity and privacy protection – Guidance on managing information security risks	This document provides guidance on implementation of the information security risk requirements specified in ISO/IEC 27001; essential references within the standards developed by ISO/IEC JTC 1/SC 27 to support information security risk management activities; actions that address risks related to information security (see ISO/IEC 27001:2022, 6.1 and Clause 8); implementation of risk management guidance in ISO 31000 in the context of information security. This document contains detailed guidance on risk management and supplements the guidance in ISO/IEC 27003. This document is intended to be used by: organizations that intend to establish and implement an information security management system (ISMS) in accordance with ISO/IEC 27001; persons that perform or are involved in information security risk management (e.g., ISMS professionals, risk owners and other interested parties); organizations that intend to improve their information security risk management process.
[7]	OIML D 34:2019 Conformity to Type (CTT) – Pre-market conformity assessment of measuring instruments	This Document provides considerations for countries and economies, or Regional Legal Metrology Organizations (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.
[8]	ISO 8601:2019	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. 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. 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.
[9]	IEEE 802.3-2022 IEEE Standard for Ethernet	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 multi-segment 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. Other specified capabilities include: various PHY types for access networks, PHYs suitable for metropolitan area network applications, and the provision of power over selected twisted pair PHY types.

Ref.	Standards and reference documents	Description
[10]	ISO/IEC 25040:2024 Information technology -- Software product evaluation	ISO/IEC 25040:2024 gives methods for measurement, assessment and evaluation of software product quality. It describes 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).
[11]	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.
[12]	WELMEC Guide 7.2, Issue 2025 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.
[13]	IEC 61131-3:2025	This document specifies the syntax and semantics of a unified suite of programming languages for programmable controllers (PCs). This suite consists of the textual language structured text (ST), and the graphical languages, ladder diagram (LD) and function block diagram (FBD). An additional set of graphical and equivalent textual elements named sequential function chart (SFC) is defined for structuring the internal organization of programs and function blocks. Also, configuration elements are defined which support the installation of programmable controller programs into programmable controller systems. In addition, features are defined which facilitate communication among programmable controllers and other components of automated systems. This edition includes the following significant technical changes with respect to the previous edition: a) inclusion of UTF-8 strings and their associated functions; b) Annex B contains a comprehensive list of features that have been added, removed or deprecated in comparison to IEC 61131-3:2013.

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 conformity with the requirements of OIML Recommendation R xyz.

The evaluation was based on OIML International Document D 31:YYYY, where the essential requirements for software are interpreted and explained. This report describes the evaluation of software needed to state conformity 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 connected transducer. 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:

Name	**Size**	**Date**	**SHA256**
main.c	12301 byte	23 Nov 2022	84dbf59a16a17e3fd4897908842b8e1a fc50fd520392b0d8770592cc82d303c3
int.c	6509 byte	23 Nov 2022	bc82f923eb2baa2608a6d646283d4b75 af56d7ad710f86f2d55356f61a7a4f84
filter.c	10897 byte	20 Oct 2022	56c049551644ebd45dff5fd7e20daf544 593b2b092ce418d095cac69c7845a88
input.c	2004 byte	20 Oct 2022	cf0f182a939977a99d00f3481e998adf2 3ba948764b53935f87f84714fe692b0
display.c	32000 byte	23 Nov 2022	93761b3938afe29867819fe407bb3956 1ae0e59c2d63e0c2825e59e7dfe22310
ethernet.c	23455 byte	15 June 2021	d99f8254aa67f8dfa8913e31321f5302 984ca162a395f73f8dbe7e0e4e721096
driver.c	11670 byte	15 June 2021	553c1c91fe147c8fee127028c3e6c983 d64673a49568779e0cf5083a4401f77c
calculate.c	6788 byte	23 Nov 2022	c4087433ec1dadcdc8e8ec6ebb05b244 594355998d7e21a7198e6c1372f3289b

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:

Name	**Identification**	**SHA256**
TT 100 User Manual	Release 1.6	799f875d6dc6b8d90f537ea9adb27ed5 c558bc845d6aaaa38feefd3d931e498a
TT 100 Maintenance Manual	Manual Release 1.1	d72f4eaf20174144a9ac9b4ac422dc89d 4ddd7b3970c2e13c15b8000e57094b0
Software description TT100	internal design document, dated 22 Nov 2022	3636421783be4ca2b304ccefa3806291 c37ac23dc6756376c6f966f1acfe363c
Electronic circuit diagram TT100	drawing no 222-31, dated 15 Oct 2022	d1a04592b42d309bbfc7f76f9ee5e271 cad765f08cffce07ca5ed5dce8abee31

The final version of the test object was delivered to the National Testing & Measurement Laboratory on 25 November 2022.

Results of evaluation

The evaluation was performed according to OIML D 31:YYYY. The evaluation was performed between 1 November and 23 December 2022. 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;
- prevention of misuse;
- indications;
- information for verification;
- software – securing and protection;
- audit trails and event counters;
- data storage;
- data transmission.

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 software module calculate.c with SDK XXX.

Result

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

6.2.2.1, 6.2.2.2, 6.2.2.3, 6.2.2.4, 6.2.2.7, 0, 6.2.3.3, 6.3.6, and 6.3.7.

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

Conclusion

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

National Testing & Measurement Lab.

Software Department

Signature(s):

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

Clause	Requirement	Passed	Failed	Remarks
6.2 General requirements				
6.2.1	The manufacturer produces measuring instruments, components and versions of the legally relevant software that conform to the approved type and the documentation submitted.			
6.2.2 Functional requirements				
6.2.2.1	Software modules of a measuring instrument or component are unambiguously, uniquely and correctly identified.			
	If the software identification consists of more than one part, at least one part is dedicated to the legal purpose.			
	The software identification is 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, the software identification is sent via a communication interface in order to be displayed or printed on another legally relevant component.			
	If the instrument facilitates remote verification, the software identification is also sent to the verification software.			
	The software identification is correctly marked on the instrument or component concerned.			
	Regardless of the form of the software identification, it is readily available when the instrument is in service to allow it to be checked.			

Clause	Requirement	Passed	Failed	Remarks
6.2.2.2	The measuring algorithms and functions of the measuring instrument are appropriate and functionally correct for the given application and device type.			
	Measures have been taken to minimize the risk of violating the MPE if a learning facility is used for dynamic modules of legally relevant software.			
	It is possible to examine algorithms and functions of the measuring instrument by metrological tests, software tests or software examination.			
6.2.2.3	The software of the measuring instrument is designed in such a way that no unreasonable demands are required from the user to obtain a correct measurement result and that the possibilities for accidental, unintentional or intentional misuse are minimal.			
6.2.2.4	The presentation of the measurement results is unambiguous for all parties affected.			
	The measurement result is displayed or printed correctly and accompanied by all measurement result relevant data necessary to inform the user of the significance of the result.			
6.2.2.5	If a display or printout is used both for legally relevant and legally non-relevant information, the legally relevant information is always readable, and clearly distinguishable from legally non-relevant information.			
6.2.2.6	If an audit trail is used, timestamps are used.			
	The timestamp is displayed in a consistent format, allowing for easy comparison of two records and tracking progress over time.			

Clause	Requirement	Passed	Failed	Remarks
6.2.2.7	It is possible to display, print and/or transmit to the verification software all necessary verification information.			
	Necessary verification information include the software identfication, current legally relevant parameter settings, data containing evidence of an intervention.			

6.2.2.8	The audit trail contains, at minimum, the following information:
	• timestamp of the event;
	• in the case of a parameter change:
	  o identification of the changed parameter;
	  o the old and new value of the changed parameter;
	• in the case of a traced update:
	  o success/failure of the update procedure;
	  o software identification of the installed version;
	  o software identification of the previously installed version;
	  o timestamp of the event;
	  o identification of the uploading party, i.e., the source of the update, if available.
	If applicable, the source of the modification is recorded in the audit trail.			

Clause	Requirement	Passed	Failed	Remarks
6.2.3 Securing and protection				
6.2.3.1	The measuring instrument is provided with the means to protect its metrological properties.			
	Software protection comprises appropriate sealing by hardware or software means, making an intervention impossible or evident.			
	In case of a software seal, a checking facility checks if no changes have occurred.			
0	Legally relevant software is secured and protected against unintentional or intentional changes and protected against accidental changes.			
6.2.3.3	Audit trail(s) and event counter(s) are part of the legally relevant software and are secured and protected as such against accidental, unintentional or intentional changes.			
	The reference number of an event counter is fixed and protected by appropriate hardware means at the time of (initial or subsequent) verification. This reference number is visibly marked on the instrument.			
	It is not possible to change the data of event counter or of an entry in an audit trail, including when software is updated.			
	It is not possible to delete an event counter or an entry in an audit trail, except as an appropriate response to free up storage capacity to allow for the addition of new entries.			

Clause	Requirement	Passed	Failed	Remarks
	Any change to the recorded data in the event counter(s) or audit trail(s), except except for deletions to free up storage capacity (as specified in 6.2.3.3), is handled as a significant software defect.			
	Events are recorded automatically, except where PGs determine specific provisions apply.			
	The audit trail(s) and event counter(s) have sufficient capacity to ensure the traceability of events between at least two successive verifications or inspections of a measuring instrument in the field .			
	If the audit trail or event counter has no more capacity, the instrument gives an appropriate response.			
6.2.3.4	Legally relevant parameters are secured and protected against accidental, unintentional or intentional changes.			
	Legally relevant parameters that require setting by the user without the need for subsequent verification are fitted with an audit trail.			
6.2.3.5	Setting the clock is secured and protected against unintentional or intentional changes.			
	Automatic setting of the time is only possible if legal time according to national legislation is used as a time base, in an authenticated manner			
6.2.3.6	During processing, measurement data are secured and protected against accidental, unintentional or intentional changes.			

Clause	Requirement	Passed	Failed	Remarks
0 Interfaces				
6.2.3.7.1	It is not possible to inadmissibly influence the legally relevant software, parameters or measurement data through protective interfaces.			
	Each command in the legally relevant software is unambiguously assigned to all functions or data changes triggered by it.			
	Only documented functions are activated through the protective interface.			
6.2.3.7.2	All inputs from the user interface are handled by a protective interface.			
6.2.3.7.3	All inputs from communication interfaces are handled by a protective interface.			
6.2.3.7.4	Hardware interfaces not equipped with a protective interface are not able to inadmissibly influence the legally relevant software, parameters, or measurement data.			
6.3 Requirements specific for configurations				
6.3.2 Detection of significant defects				
6.3.2.2	If software is involved in the detection of significant defects, it performs such checks at regular intervals.			
	If software is involved in the detection of significant defects, it responds appropriately to any detected defect.			

Clause	Requirement	Passed	Failed	Remarks
6.3.3 Detection of durability errors and/or significant faults				
6.3.3.2	If software is involved in durability protection or the detection of significant faults, it performs such checks at regular intervals.			
	If software is involved in durability protection or the detection of significant faults, it responds appropriately to any detected durability error or significant fault.			
6.3.3.3	If an error log is used to fulfill requirements from 6.3.3.2, error logs are part of the legally relevant software and are secured and protected as such against accidental, unintentional or intentional changes.			
	It is not possible to change the data of an entry in an error log, including when software is updated.			
	It is not possible to delete an entry in an error log, except as an appropriate response to free up storage capacity to allow for the addition of new entries.			
	Entries in an error log are recorded automatically.			
6.3.4 Dynamic modules of legally relevant software				
6.3.4.1	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 includes information regarding the use of those software modules in the measurement process.			

Clause	Requirement	Passed	Failed	Remarks
6.3.4.2	Measuring functions are not inhibited nor affected by a continuous learning process.			
	It is not possible to make any modifications to parameters during a measurement.			
	Changes of predefined parameters within dynamic modules of legally relevant software are protected.			
0 **Compatibility of operating systems and** hardware				
6.3.5.2 Functional requirements				
6.3.5.2.1	The configuration of the operating system is made identifiable as described in 6.2.2.1.			
	The identifier is displayed on command or during operation and, if applicable, transmitted to the verification software by the measuring instrument.			
6.3.5.3 Securing and protection				
6.3.5.3.1	Legally relevant configuration settings of the operating system are protected.			
	The administration tasks of the legally relevant software are protected.			
6.3.5.3.2	The access control feature of the operating system is configured in such a way that the intended use cannot be inadmissibly influenced.			
6.3.5.3.4	The boot process ensures integrity and authenticity of the legally relevant software.			
	The boot configuration is secured and protected.			

Clause	Requirement	Passed	Failed	Remarks
	Booting via open interfaces is disabled.			
6.3.5.3.5	Communication with the legally relevant software is made via protective interfaces.			
6.3.5.3.6	Insufficient resources or an unsuitable environment cannot inadmissibly influence the measurement result.			
	If insufficient resources or an unsuitable environment are detected by the instrument, it responds appropriately			
6.3.6 Data storage				
6.3.6.2.1	The stored measurement data include all relevant data necessary for future legally relevant use.			
6.3.6.2.2	Data are stored automatically.			
	A checking facility regularly checks the availability of the storage and in the case the storage device is not available or full, this is handled accordingly.			
	When the measurement data necessary for the calculation of the measurement result are relevant for legal purposes, all measurement result relevant data included in the calculation are automatically stored with the final value.			
	Measurement data stored in a component to construct the measurement result are only deleted if the next software module or component has checked and stated a proper completion of all expected actions.			

Clause	Requirement	Passed	Failed	Remarks
6.3.6.2.3	The measurement result is only deleted if the transaction is settled, or if these data are printed by a printing device subject to legal control.			
6.3.6.3	The stored data are protected against accidental, unintentional, or intentional changes.			
	If appropriate, means are provided whereby cryptographic keys can only be input or read if a hardware seal is broken.			
	The software that displays, or further processes, the measurement data checks the authenticity and integrity of the data after having read them from the storage.			
	If an irregularity is detected, the software responds appropriately.			
6.3.7 Data transmission				
6.3.7.2	The transmitted measurement data include all data necessary for future legally relevant use.			
6.3.7.3	The transmitted data are protected by software means to guarantee authenticity and integrity.			
	If appropriate, means are provided whereby cryptographic keys used by cryptographic methods can only be input or read if a seal is broken.			
	Software modules that prepare measurement data for sending or that check measurement data after receiving are part of the legally relevant software.			

Clause	Requirement	Passed	Failed	Remarks
	The software that displays, or further processes, the measurement data checks authenticity and integrity of the data received from a transmission channel.			
	If an irregularity is detected, an appropriate response is given.			
6.3.7.4	The measurement cannot be inadmissibly influenced by a transmission delay or by the interruption or unavailability of network services or this can be detected in which case an appropriate action is required.			
	If a transmission delay or the interruption or unavailability of network services occurs, an appropriate response is given.			
6.3.8.2 Specification and separation of components				
6.3.8.2.2	If a component is shared by multiple components, e.g., one display for multiple sensors, then all the components that share another component are unambiguously identified.			
6.3.8.2.3	All legally relevant components are 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 secured and protected.			
	Legally relevant components check the authenticity, integrity and/or availability of other software-controlled components. When the authenticity and/or integrity check fails, or the other component is not available, the checking component responds appropriately.			

Clause	Requirement	Passed	Failed	Remarks
	Legally non-relevant components or devices are prevented from calculating/presenting/spoofing the measurement result.			
	If legally relevant components have limited functionality and limited securing/protection capabilities, they have limited access to the measurement data, i.e., they only indicate the measurement data without modification.			
	The measurement data are prepared for transmission or storage for further processing by a component that can be fully secured and protected.			
	The receiving component is capable of checking the authenticity and integrity of the measurement data.			
	If increased protection against fraud is necessary, a component exists with increased securing means that can display or print the measurement results in case of a dispute.			
6.3.8.3 Specification and separation of software modules				
6.3.8.3.1	If the separation of the software is not possible or needed, the software is legally relevant as a whole.			
6.3.8.3.2	If the manufacturer chooses a mixed identifier for legally relevant and legally non-relevant software, the legally relevant software identifier(s) is/are clearly distinguishable from the non- legally relevant part.			
6.3.8.3.3	Legally non-relevant software modules are prevented from calculating/presenting/spoofing the measurement result.			

Clause	Requirement	Passed	Failed	Remarks
	All legally relevant software modules communicate with other software modules and/or components through a protective interface			
	The legally relevant process is not inadmissibly interrupted by legally non-relevant software.			
	The measurement process (realized by the legally relevant software) is not delayed or blocked by other processes.			
6.3.9 Maintenance and reconfiguration				
6.3.9.2	An update does not inadmissibly influence the measurement process.			
6.3.9.3 Verified update				
6.3.9.3.2	Access to the verified update is protected by a hardware or software seal that must be broken for the update to take effect.			
6.3.9.4 Traced update				
6.3.9.4.2	After initiation of the update procedure, the traced update of software runs automatically.			
	If a feature is required for the user or owner to express their consent prior to an update, it is possible to enable and disable the feature.			
	If the user or owner denies consent, the update procedure does not start at all.			
	The software update is recorded in an audit trail.			
	A traced update does not affect existing parameters.			

Clause	Requirement	Passed	Failed	Remarks
0	A traced update does not influence the legally relevant parameters.			
	If some of the securing or protection means of the instrument are turned off to enable updating, they are turned on again automatically immediately after update, regardless of the result of the update process.			
	During or as a result of a traced update, any existing protection means, e.g., audit trail information and event counter values, are retained.			
6.3.9.4.4	Technical means are employed to guarantee the authenticity of the loaded software, i.e., that it originates from the owner of the certificate.			
	Technical means are employed to ensure the integrity of the loaded software, i.e., that it has not been inadmissibly changed before loading.			
	If the loaded software fails the integrity test or the authenticity test, the instrument discards the new version and uses the previous version of the software or switches to an inoperable mode. In this mode, measuring functions are inhibited. Only two things are possible: Either the user can manually restart the download procedure or the instrument displays an error message.			
0	The audit trail contains, at minimum, the following information: success/failure of the update procedure; software identification of the installed version; software identification of the previously installed version; timestamp of the event; identification of the uploading party, i.e., the source of the update, if available.			

Clause	Requirement	Passed	Failed	Remarks
	The storage device that supports the traced update has sufficient capacity to ensure the traceability of traced updates of the legally relevant software between at least two successive verifications or inspections of a measuring instrument in the field.			
	If the audit trail has no more capacity, an appropriate response is given.			
6.3.10 Remote verification capability				
6.3.10.1	In case the instrument facilitates remote verification, the requirements in 6.3.10.2 to and 6.3.10.3 are met.			
6.3.10.2 Functional requirements				
6.3.10.2.1	For the purpose of remote verification, the instrument uses timestamps, provides evidence of an intervention, uses audit trails, stores logging data, has a facility for detection of significant defects and makes these available for remote verification purposes.			
	There is a legally relevant interface for data extraction for remote verification purposes.			
	It is possible to establish and ensure the integrity of the instrument to be verified.			
	When checking software integrity, the integrity measure (checksum, hash) is calculated immediately before transmitting the integrity measure to the remote verification software.			
	It is possible to establish the authenticity of the instrument, i.e., the instrument is uniquely identified, and other means are provided to ensure authenticity.			

Clause	Requirement	Passed	Failed	Remarks
	The remote verification functionality is secured.			
6.3.10.2.2	Test items are uniquely identified. The obtained test items are unambiguously linked to the measuring instrument to be verified.			
	Relevant test items are available depending on the specific requirement to be tested and the instrument type .			
6.3.10.2.3	The result of the remote verification contains at least a unique ID (at least identifying the verification authority) and the date of the verification.			
6.3.10.3	Interfaces for remote verification are protected.			
	The connection to the remote verification software complies with 6.3.7.			
	The software modules involved in the remote verification procedure are part of the legally relevant software and fulfill the relevant requirements.			
	An ongoing measurement is not influenced by remote verification.			
	The use of the verification procedure does not influence the compliance with other requirements.			
	The software integrity of the instrument is not influenced by the remote verification procedure.			
	The access to the verification procedures, specific test items or commands are restricted if these influence compliance with other requirements			

Clause	Requirement	Passed	Failed	Remarks
	Provisions are made to securely store the result of the remote verification in the measuring instrument. These data are protected and secured.			
	Stored results of the verification in the instrument comply with 6.3.6.			
	Securing ensures that only the remote verification software has write permissions.			
	The oldes entry of records can only be deleted after the minimum storage period for results of a remote verification has elapsed and if the storage device has no more capacity.			

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.

In this Document, the definition of *Measurement Result (3.2.42)* is a "set of quantity values being attributed to a measurand together with any other relevant data", (i.e., Measurement Result Relevant Data). This is illustrated in Figure A.1 as the Measured Quantity Value (MQV) and Measurement Result Relevant Data (MRRD), both being part of the Measurement Result.

Together with the Measurement Process Data (MPD) these form the Measurement Data.

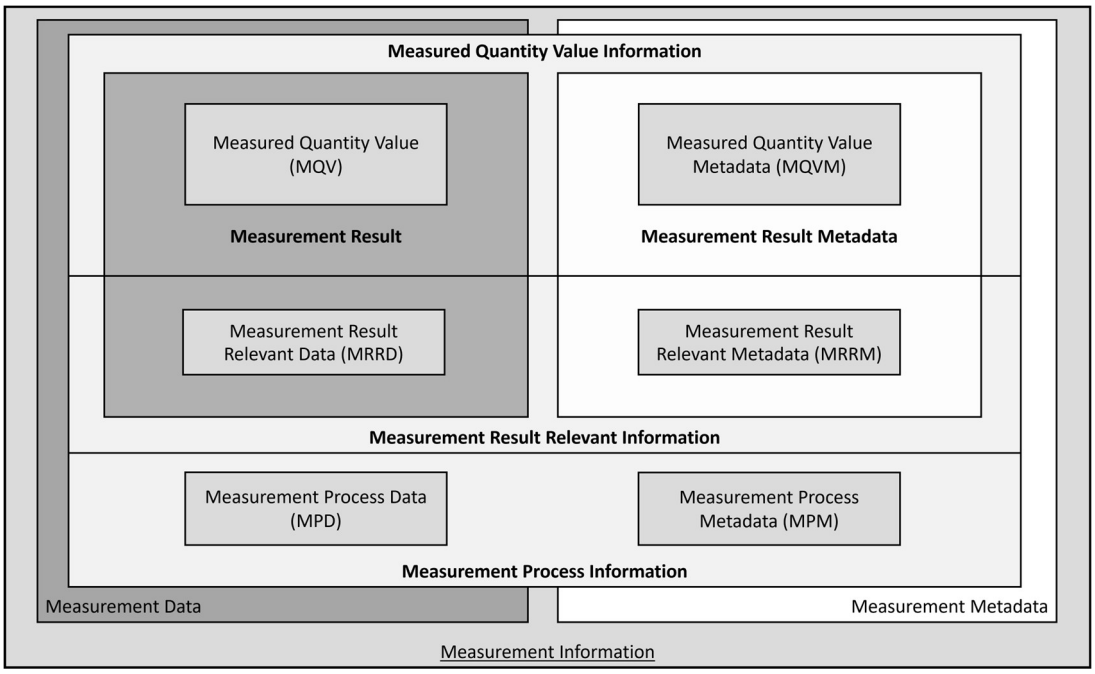

Figure A.1 – Visual representation of the Measurement Information

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.

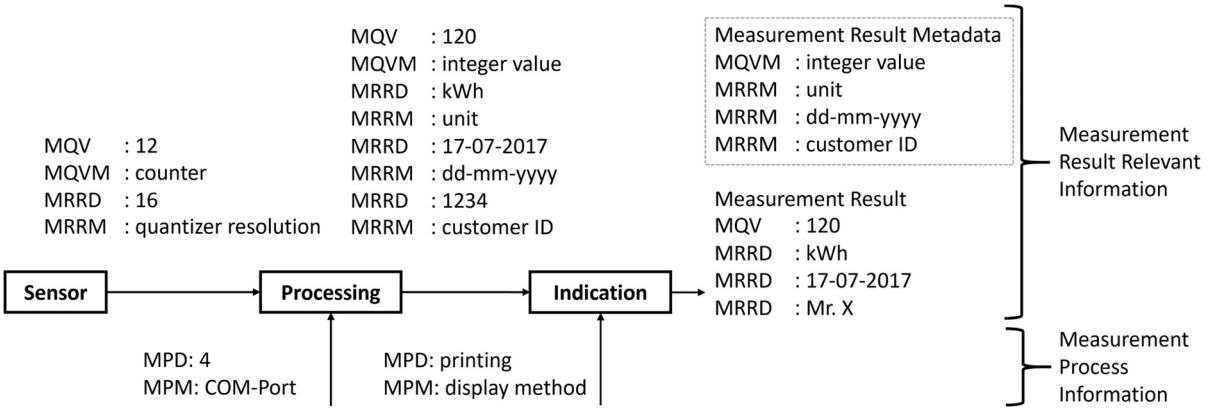

Figure A.2 – Flowchart of a measurement process, giving examples for the different data relevant to the Measurement Result or relevant do the Measurement Process.

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 are 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 (MRRI) are 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 Value Metadata (MQVM) is assigned ‘kWh’ as Measurement Result Relevant Data (MRRD) with ‘unit’ as Measurement Result Relevant Metadata (MRRM), as well as a timestamp ‘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 are 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 are 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
How to adopt requirements in an OIML Recommendation (Informative)

This Annex provides information for PGs on how to implement software requirements in a Recommendation based on this Document.

To facilitate the adoption of the requirements in an OIML Recommendation, additional information is provided in the main text, apart from the actual requirements.

Notes and examples are included in the text to provide additional information for understanding the listed requirement. PGs may decide to refer to this Document for these notes and examples and to include instrument specific notes and examples in the Recommendation where necessary.

Where a solution to meet a specific requirement needs to be documented or stated in the certificate, this is marked with the labels “documentation” and “certificate”. The documentation requirements are summarized in clause 7.1, the information to be included in the certificate can be found in clause 7.2. It is up to the PGs to decide to include this additional text or only keep 7.1 and 7.2.

The requirements in this Document are of a general nature. Where a PG needs to adopt a specific general requirement for a particular type of instrument, this is indicated by the label “Guidance”. Guidance is directed to the PG to take actions or make decisions to adopt the requirement.

Additionally, a PG has to make a selection of specific technical requirements for certain technologies that are currently present in the type of measuring instrument or is anticipated to be part of measuring instruments to be regulated by the Recommendation in the near future.

The guidance for PGs uses the normative verbs “may” and “should”. “May” signifies that guidance is optional and the requirement can stand on its own. “Should” implies that PGs have to follow the guidance because the requirement is incomplete otherwise.

The documentation guidance uses the normative verb “shall” to indicate that including the specified information in the documentation is mandatory.

The first part of this Annex lists general actions and decisions any PG shall make when adopting the requirements of this Document, labelled in the main text with Guidance.

The second part provides instructions regarding implementation of the individual clauses of this Document.

It is recommended to start implementing the requirements of this Document in this order.

PG actions and decisions

The following list summarizes decisions to be made by PGs before implementing the requirements of this Document in a Recommendation.

Clause	Decision
4.3	This Document lists two risk levels, normal risk level and raised risk level. PGs should determine which risk level is suitable. In clause 5, some aid is given for performing this task.
4.3	PGs should decide which metrological characteristics (at least legally relevant software, parameters and measurement data) shall comply with the requirements.
4.4	PGs should decide which parameters are legally relevant for a specific application.
4.4	PGs should decide which measurement data are legally relevant and shall comply with the requirements, see Annex C, and PGs should also decide which metadata shall be documented by the manufacturer.
6.2.2.1	PGs may decide which forms of software identification are permissible.
6.2.2.1	As an exception, an imprint of the software identification on the measuring instrument or component can be an acceptable solution. PGs should allow or disallow this exception.
6.2.2.4	PGs should specify the measurement result relevant data that need to be indicated.
6.2.2.4	PGs should specify the layout of the display and printout for the legally relevant information.
6.2.2.4	PGs may also specify the requirements for the display and/or printout of the legally relevant information.
6.2.2.6	PGs may define minimum requirements for the resolution of timestamps to ensure consistency and applicability.
6.2.2.7	PGs may define what verification information is necessary for the instrument type.
6.2.2.8	Audit trails shall contain, at minimum, certain information. PGs may define additional information to be recorded in the audit trail. This could be the source of the modification, details of a verified update or an indication that the modification is the result of an action of dynamic modules of legally relevant software or of remote verification.
6.2.3.3	PGs should define for specific types of instruments which manual additions to an event in the audit trail are admissible, as long as they do not affect the remaining contents of the audit trail.
6.2.3.3	PGs should specify the capacity required for the audit trail and event counter and the response required, i.e., either the oldest entry may be deleted, or no other change of a parameter shall be possible without breaking a seal, or the event counter may restart the numbering.
6.2.3.5	Setting the clock shall be secured and protected against accidental, unintentional or intentional changes. PGs may decide to exempt certain types of measuring instruments from this requirement.
6.2.3.5	PGs may define requirements and test methods for internal clocks in cases where accurate time is required for a legally relevant purpose.
6.3.2.1	PGs may require functions to detect significant defects, noting that in case of a software implemented seal a checking facility is required to check for changes, see 6.3.2.2. 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.
6.3.2.2	PGs should determine which interval is required for the checks for significant defects.
6.3.2.2	PGs should prescribe an appropriate response, e.g., that the instrument or component is deactivated or an alarm/record in an error log is generated if a significant defect is detected.
6.3.3.1	PGs may require functions to detect durability errors and significant faults. In this case, the manufacturer of the measuring instrument shall be required to design detection functions into the

Clause	Decision
	software modules or hardware components or provide means by which the hardware components can be supported by the software modules of the instrument.
6.3.3.2	PGs should determine which interval is required for the checks for durability errors and significant faults.
6.3.3.2	PGs should prescribe an appropriate response, e.g., that the instrument or component is deactivated or an alarm and/or record in an error log is generated if durability is detected as being jeopardized or a significant fault is detected.
6.3.4.1	The documentation of the software functions shall include a detailed description of several aspects of the dynamic module. PGs may decide not to implement this requirement in their Recommendation.
6.3.4.2	PGs should decide if a subsequent verification is required when a legally relevant parameter is changed by the dynamic modules of legally relevant software. To allow for the possibility of parameter adaptations in dynamic modules of legally relevant software without subsequent verification, the source of the parameter change (e.g., the learning facility) is logged in the audit trail, see 6.2.3.3.
6.3.5.3.6	Insufficient resources or an unsuitable environment shall not inadmissibly influence the measurement result. If insufficient resources or an unsuitable environment are detected by the instrument, it shall respond appropriately, see 6.3.2. PGs should consider fixing the hardware, operating system, or system configuration of a universal device or even excluding the usage of an off-the-shelf universal device in the following cases: • if there is a raised risk level; • if cryptographic algorithms or keys need to be implemented (see 6.3.6 and 6.3.7).
6.3.6.1	For different applications, PGs may decide if storage of measurement data is required and if additional data need to be stored.
6.3.6.2.1	PGs should decide which measurement data, e.g., measurement result relevant data necessary to reconstruct the measurement result, shall be stored.
6.3.6.2.2	PGs should decide which measurement data are relevant for legal purposes.
6.3.6.2.3	PGs should decide how long records that store results of a remote verification shall be kept.
6.3.6.2.3	PGs may define alternative conditions for data deletion.
6.3.6.3	PGs may require a raised risk level when considering a freely accessible storage, i.e., storage that is accessible without violating securing and protection means.
6.3.6.3	PGs may specify appropriate responses to detected irregularities in stored data, e.g., the data shall be discarded or marked unusable.
6.3.6.3	PGs may set limitations on storage solutions, e.g., whether or not data shall be stored locally, in different locations or in the cloud.
6.3.7.2	PGs should decide which measurement data (e.g., measurement result relevant data necessary to reconstruct the measurement result) shall be transmitted.
6.3.7.3	PGs may require a raised risk level when considering a publicly accessible open network. Raised risk levels might require application of cryptographic methods.
6.3.7.3	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 response shall be required. PGs should decide what response is required, e.g., the measurement data shall be discarded or marked unusable.

Clause	Decision
6.3.7.4	The measurement shall not be inadmissibly influenced by a transmission delay, or interruption or unavailability of network services or this shall be detected in which case an appropriate response shall be required. PGs should decide what response is required, e.g., disabling the measurement function, stopping the current measurement process, discarding the measurement result or marking it as unusable.
6.3.8.1	PGs may specify the software modules, components or parts of the software modules or components that are legally relevant.
6.3.8.2.2	PGs should decide if it is always required to identify components on a print-out. This could be relevant in case where the product bears a label or the measurement is repeatable.
6.3.8.2.3	Legally relevant components shall be protected against exchange. PGs may decide to exempt some components from this requirement, e.g., in the case of simple recipient printers.
6.3.8.2.3	Legally relevant components shall check the authenticity, integrity and/or availability of another software-controlled component. PGs should decide which action shall be taken if the authenticity and/or integrity check fails.
6.3.8.2.3	PGs may decide that certain components shall be connected and available on site, for example a display or a printer.
6.3.8.2.3	Legally relevant components shall check the authenticity, integrity and/or availability of another software-controlled component. PGs may decide to exempt some components from this requirement, e.g., in the case of simple recipient printers it could be that only availability needs to be checked.
6.3.9.1	PGs should decide whether a verified or a traced update is allowed.
6.3.9.3.2	After the update of the legally relevant software of a measuring instrument (exchange with another approved software version or re-installation), the securing and protection means should be renewed and the measuring instrument should be verified. PGs may also specify other procedures following a verified update.
6.3.9.4.1	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.
6.3.9.4.2	Traced update: PGs should decide if it is necessary for the user or owner to express their consent prior to an update, e.g., by means of a push button.
0	In case of a traced update, PGs should specify a sufficient capacity for the audit trail and the required response, i.e., either the oldest entry may be deleted or the update procedure should not start at all.
6.3.10.2.2	In case of remote verification, PGs should define a list of relevant test items for verification purposes, e.g., approved type number, serial number, legally relevant settings and parameters, verification information and status, software identification, software integrity, audit trails, error logs etc.
6.3.10.2.3	The result of the remote verification shall contain, at least, a unique ID (at least identifying the verification authority) and the date of the verification. PGs should decide which additional data shall be stored.
6.3.10.3	The Access to the verification procedures, specific test items or commands shall be restricted if these influence compliance with other requirements, such as requirements on battery life, on resources, or delays in the measurement process. PGs should decide if access to the verification procedure shall always be restricted.

Clause	Decision
7.1.1	Regarding 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.
7.4	PGs should justify the selection of level B 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 approved type, and risk of wrong measurement result due to operating errors.
8.1	PGs should consider the subclause 8.2 when writing instrument-specific verification procedures. The methods given in 8.2 are proposed as the standard procedure.
8.3.2	PGs should define a list of relevant data for verification purposes depending on the instrument type (approved type number, serial number, legally relevant settings and parameters, verification information and status, software identification, software integrity, audit trails, error logs etc.).
8.3.2	PGs should select the appropriate remote verification procedures depending on the type of instrument. Instrument-specific verification procedures (see 8.3.3.3) shall be detailed in the relevant Recommendation.

Elements to be implemented in a Recommendation

In the following, adaptation instructions are provided for specific clauses such as clause 3, 6, 7 and 8. This also addresses selection of specific technical requirements for certain technologies from clause 6.

Clause 3 "Terms and definitions"

Terms and definitions from clause 3 of this Document should only be copied to a Recommendation if they are needed for understanding related requirements. Whenever possible, PGs should consider referencing the terms and definitions of this Document instead to avoid conflicting implementations.

Clause 6 "Requirements for measuring instruments regarding software"

General requirements from clause 6.2 should, in principle, be applicable to all types of measuring instruments and should be copied to an OIML Recommendation as a baseline for software requirements.

Requirements from clause 6.3 for specific configurations should only be copied if the individual configuration is legally required or currently present in the type of measuring instrument or is anticipated to be part of measuring instruments to be regulated by the Recommendation in the near future.

Clause 7 "Type evaluation"

Documentation requirements from clause 7.1 should be copied to the respective clause in part 1 of any Recommendation. PGs should pay special attention to restricting documentation requirements to those related to the requirements implemented from clause 6, see above.

Information to be included in the certificate from clause 7.2.2 should also be copied to the respective clause in part 1 of the Recommendation under development. PGs should be aware that only those information are needed, for which corresponding software requirements have been implemented.

Clause 7.3 will usually be integrated into parts 3 (test methods) and 4 (verification methods) of a Recommendation. PGs should only copy those test and verification methods applicable for the selected risk level.

Note 1: The evaluation and verification methods for examination levels A and B shown in Table 2 in clause 7.3.1 only constitute Recommendations. While this Document strongly recommends using these evaluation and verification methods, PGs may make a different selection or even add verification and examination methods where needed.

Note 2: Clauses 6.3.2.1 and 6.3.3.1 are optional clauses dependent on PG decisions, see above. If a PG decides not to include any one these clauses, the corresponding row in Table 2 shall be deleted.

Clause 8 Verification of a measuring instrument

Clause 8.2 will typically be included in part 4 (verification methods) of a Recommendation. It is assumed that the aspects of 8.2 will be applicable to all types of measuring instruments and can be copied directly.

If a PG decides to implement requirements for partial or full remote verification (see the list under "PG actions and decisions" in this Annex), clause 8.3 shall be used to draft corresponding clauses in part 4 of the Recommendation under development. It is recommended to copy clauses 8.3.3.1 and 8.3.3.2 directly to the Recommendation since they should be applicable to all types of measuring instruments. If applicable, specific examples from clause 8.3.3.3 may also be copied. Otherwise, 8.3.3.3 may serve as a basis for formulating other instrument-specific remote verification procedures.

Annex E
Comparison table

OIML D 31:2023		OIML D 31 2.1CD		Remarks
Ref.	**Description**	**Ref.**	**Description**	
3.2	General terminology	3.2	General terminology	A definition for learning facility has been added.
4	Instructions for use of this Document in drafting OIML Recommendations	4	Instructions for use of this Document in drafting OIML Recommendations	Clause 4 has been completely rewritten to reflect the addition of the new informative Annex D. Applicability to OIML Guides has been added.
5.2	Selection of risk levels	5.2	Selection of risk levels	Duplicate description of risk levels (also contained in 6.1) has been deleted. The explanation of the connection between risk levels and examination levels has been rephrased.
6	Requirements for measuring instruments with respect to software	6	Requirements for measuring instruments regarding software	Clause 6 has been reordered completely. Requirements are now sorted according to the individual instrument property (software, parameters, data etc.). In addition, requirements have been split into functional and protection/securing requirements.
6.2.6.1	Detection of significant defects	6.3.2	Detection of significant defects	Detection of faults has been integrated into clause 6.3.3, see also modified definition

				3.2.55.
6.2.6.2	Durability protection	6.3.3	Detection of durability errors and/or significant faults	Detection of faults has been moved here from 6.2.6.1.
6.3.4.4.2	Requirements for deletion of measurement data	6.3.6.2.3	Deletion of the stored measurement result	Deletion of the measurement result has been turned into a separate subclause.
6.3.8	Figure – software update procedure	6.3.11	Figure – software update procedure	Figure has been updated: References to clauses and notes have been removed.
6.3.8.4	Traced updated	6.3.9.4	Traced updated	Clause has been reordered to mirror the procedure in the field.
7.1.2	Contents of the documentation	7.1.2	Contents of the documentation	Updated to correspond to the revised Documents
7.3.1	Table 2 – Recommendations for combinations of evaluation and verification methods	7.3.1	Table 2 – Recommendations for combinations of evaluation and verification methods	The table has been updated to reflect changes throughout the document.
Annex B	Example of a software test report	Annex B	Example of a software test report	The report template has been updated to reflect changes in clause 6.
-	-	Annex D	How to adopt requirements in an OIML Recommendation	A new informative annex has been added to help PGs when adopting the requirements of this Document in a Recommendation and to summarize all decisions PGs have to make during adoption.
-	-	Annex E	Comparison Table	This comparison table has been added as a separate annex to

				highlight major changes relative to D31:2023.
Annex D	Index	Annex F	Index	The index has been updated to reflect changes throughout the document.

Annex F

Index

Audit trail: 3.2.1; 3.2.57; 6.2.2.6; 6.2.2.7; 6.2.2.8; 6.2.3.1; 6.2.3.3; 6.2.3.36.2.3.4; 6.3.2.2; 6.3.4.2; 6.3.5.3.1; 6.3.8.2.3; 6.3.8.3.1; 6.3.9.4.2; 0; 6.3.10.2.1; 7.1.2; 7.2.2; 7.3.2.3; 8.2.3.2; 8.2.4; 8.3.2; 8.3.3.1; 8.3.3.2.2; 8.3.3.2.4; Annex B; Annex D.

Authentication: 3.2.2; 3.2.3; 6.3.5.3.4; 6.3.9.4.2.

Authenticity: 3.2.3; 3.2.10; 3.2.14; 6.2.3.1; 6.2.3.5; 6.3.5.3.4; 6.3.6.3; 6.3.7.3; 6.3.8.2.3; 0; 6.3.10.2.1; 8.3.2; Annex B; Annex D.

Checking facility: 3.2.5; 6.2.3.1; 6.3.2.1; 6.3.2.2; 6.3.6.2.2; Annex B; Annex D.

Command: 3.2.52; 3.2.62; 6.2.2.1; 6.2.2.3; 6.2.3.7.1; 6.2.3.7.2; 6.3.5.2.1; 6.3.5.3.5; 6.3.8.2.3; 6.3.10.3; 7.1.1; 7.1.2; 7.2.1; 7.3.1; 7.3.2.1; 7.3.2.3; 7.3.2.4; 8.3.1; Annex B; Annex D.

Communication: 3.2.7; 3.2.71; 5.2; 6.2.2.1; 6.2.3.7.3; 6.3.5.3.5; 6.3.5.3.6; 6.3.7.4; 6.3.8.2.3; 6.3.8.3.1; 6.3.8.3.3; 6.3.10.2.1; 7.2.2; 7.3.1; 7.3.2.1; 8.3.2; Annex A; Annex B.

Communication interface: 3.2.7; 5.2; 6.2.2.1; 6.2.3.7.3; 7.3.1; Annex B.

Component: 2.3; 3.2.7; 3.2.8; 3.2.12; 3.2.22; 3.2.31; 3.2.32; 3.2.64; 3.2.73; 3.2.75; 6.1; 6.2.1; 6.2.2.1; 6.2.2.5; 0; 6.3.2.1; 6.3.2.2; 6.3.3.1; 6.3.3.2; 6.3.6.2.2; 6.3.6.3; 6.3.7.3; 6.3.7.4; 6.3.8.1; 6.3.8.2.1; 6.3.8.2.2; 6.3.8.2.3; 6.3.8.3.1; 6.3.8.3.3; 6.3.9.1; 7.1.2; 7.3.1; 7.5; Annex A; Annex B; Annex D.

Cryptographic certificate: 3.2.9; 3.2.14; 6.2.3.1; 0; 6.2.3.5; 6.3.8.2.3.

Cryptographic means: 3.2.10; 6.2.3.1; 6.3.9.4.4; Annex B.

Data domain: 3.2.11; 3.2.52; 3.2.62; 3.2.63; 3.2.64; 6.3.6.2.2; 6.3.8.3.3; 7.1.2; 7.3.2.4.

Device-specific parameter: 3.2.12; 3.2.16; 3.2.31; 6.2.3.4; 6.3.9.1; 8.1.

Digital Signature: 3.2.9; 3.2.10; 3.2.14; 6.2.3.1; 0; 6.3.6.3; 6.3.7.3; 6.3.8.2.3; 6.3.9.4.4.

Durability: 3.2.15; 6.3.3.1; 6.3.3.2; 7.1.2; 7.3.1; 8.3.3.3.3; Annex B; Annex D.

Dynamic module of legally relevant software: 3.2.16; 3.2.58; 6.2.1; 6.2.2.8; 6.3.4.1; 6.3.4.2; 6.3.5.3.6; 6.3.6.2.1; 6.3.7.2; 6.3.8.3.1; 6.3.8.3.3; 7.1.1; 7.1.2; 7.2.2; 7.3.1; 0; 7.3.2.5; 8.1; Annex B; Annex D.

Electronic measuring instrument: 3.2.17; 3.2.23; 6.3.8.3.3.

Error (of indication): 3.2.18; 3.2.23; 3.2.28; 3.2.33.

Error log: 3.2.19; 6.3.2.2; 6.3.3.2; 6.3.3.3; 6.3.7.1; 6.3.10.2.2; 8.3.2; 8.3.3.1; Annex D.

Evaluation (software): 7.1.1; 7.1.2; 7.2.1; 7.3.1; 7.3.2.1; 0; 7.3.2.3; 7.4; 8.3.1; Annex A; Annex B.

Evaluation (type): 3.2.50; 3.2.72; 3.2.73; 3.2.77; 6.2.1; 6.2.3.7.2; 6.3.8.2.1; 6.3.8.3.3; 7.1.1; 7.1.2; 7.2.1; Annex D.

Event: 3.2.1; 3.2.20; 3.2.21; 3.2.64; 3.2.69; 6.2.2.6; 6.2.2.7; 6.2.2.8; 6.2.3.1; 6.2.3.3; 6.2.3.4; 6.2.3.7.4; 6.3.8.2.3; 0; Annex B; Annex D.

Event counter: 3.2.21; 6.2.2.7; 6.2.3.3; 6.2.3.4; 0; 7.2.2; 8.3.3.1; 8.3.3.2.4; Annex B; Annex D.

Executable code: 3.2.22; 3.2.66; 6.2.2.1; 6.3.10.2.1; Annex B.

Fault: 3.2.23; 3.2.57; 6.3.2.1; 7.1.2; 7.3.2.3; Annex B; Annex D.

Hardware protection: 3.2.55; 5.2; **Error! Reference source not found.**; 6.3.3.2; 6.2.3.4; 6.3.5.3.4; 6.3.6.3; 7.2.2; 8.3.1.

Hash function: 3.2.24; 6.3.2.2.

Integrity (of programs, data, or parameters): 3.2.10; 3.2.14; 3.2.25; 3.2.59; 3.2.68; 6.3.5.3.4; 6.3.6.3; 6.3.7.3; 6.3.8.2.3; 6.3.10.2.1; 6.3.10.2.2; 6.3.10.3; 7.2.2; 8.1; 8.2.2; 8.2.3.2; 8.3.1; 8.3.2; 8.3.3.1; 8.3.3.2.1; 8.3.3.2.3; 8.3.3.2.4; Annex A; Annex B; Annex D.

Interface: 3.2.7; 3.2.26; 3.2.52; 3.2.62; 3.2.64; 3.2.65; 3.2.75; 5.2; 6.2.2.1; 6.2.2.3; 6.2.2.5; 6.2.3.6; 6.2.3.7.1; 6.2.3.7.2; 6.2.3.7.3; 6.2.3.7.4; 6.3.5.1; 6.3.5.2.1; 6.3.5.3.4; 6.3.5.3.5; 6.3.8.2.1; 6.3.8.2.3; 6.3.8.3.3; 6.3.10.2.1; 6.3.10.3; 7.1.1; 7.1.2; 7.3.1; 7.3.2.1; 7.3.2.3; 7.3.2.4; 8.3.1; Annex A; Annex B.

Legally relevant: 2.1; 3.2.1; 3.2.12; 3.2.16; 3.2.19; 3.2.20; 3.2.30; 3.2.31; 3.2.32; 3.2.42; 3.2.52; 3.2.58; 3.2.59; 3.2.62; 3.2.65; 3.2.73; 3.2.74; 4.3; 4.4; 6.2.1; 6.2.2.1; 6.2.2.2; 6.2.2.3; 6.2.2.4; 6.2.2.5; 6.2.2.6; 6.2.2.7; 6.2.2.8; 6.2.3.1; 0; 6.2.3.3; 6.2.3.4; 6.2.3.5; 6.2.3.6; 6.2.3.7.1; 6.2.3.7.2; 6.2.3.7.4; 6.3.2.2; 6.3.4.1; 6.3.4.2; 6.3.5.1; 6.3.5.2.1; 6.3.5.3.1; 6.3.5.3.4; 6.3.5.3.5; 6.3.5.3.6; 6.3.6.2.1; 6.3.6.2.2; 6.3.6.3; 6.3.7.2; 6.3.7.3; 6.3.8.1; 6.3.8.2.1; 6.3.8.2.3; 6.3.8.3.1; 6.3.8.3.2; 6.3.8.3.3; 6.3.9.1; 6.3.9.3.2; 6.3.9.4.1; 0; 6.3.10.2.1; 6.3.10.2.2; 6.3.10.3; 7.1.1; 7.1.2; 7.2.2; 7.3.1; 7.3.2.1; 0; 7.3.2.5; 8.1; 8.3.2; 8.3.3.1; 8.3.3.2.2; Annex B; Annex D.

Legally relevant parameter: 3.2.12; 3.2.20; 3.2.31; 3.2.73; 6.2.2.3; 6.2.2.7; 6.2.3.4; 6.2.3.5; 6.3.2.2; 6.3.4.2; 0; 7.1.2; Annex B; Annex D.

Legally relevant software: 2.1; 3.2.16; 3.2.20; 3.2.32; 3.2.42; 3.2.52; 3.2.58; 3.2.59; 3.2.65; 3.2.73; 4.3; 6.2.1; 6.2.2.5; 6.2.2.6; 6.2.2.8; 0; 6.2.3.6; 6.2.3.7.1; 6.2.3.7.4; 6.3.2.2; 6.3.3.3; 6.3.4.1; 6.3.4.2; 6.3.5.3.1; 6.3.5.3.4; 6.3.5.3.5; 6.3.5.3.6; 6.3.6.2.1; 6.3.6.3; 6.3.7.2; 6.3.7.3; 6.3.8.2.3; 6.3.8.3.1; 6.3.8.3.2; 6.3.8.3.3; 6.3.9.3.2; 0; 6.3.10.2.1; 6.3.10.3; 7.1.1; 7.1.2; 7.2.2; 7.3.1; 0; 7.3.2.5; 8.1; 8.3.2; 8.3.3.1; 8.3.3.2.2; Annex B; Annex D.

Maximum permissible error: 3.2.33; 3.3; 0.

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; 3.2.17; 3.2.19; 3.2.20; 3.2.22; 3.2.23; 3.2.30; 3.2.31; 3.2.32; 3.2.33; 3.2.34; 3.2.44; 3.2.45; 3.2.50; 3.2.54; 3.2.55; 3.2.57; 3.2.59; 3.2.61; 3.2.63; 3.2.64; 3.2.65; 3.2.72; 3.2.73; 3.2.75; 3.2.77; 3.2.78; 4.3; 5.1; 5.2; 6.1; 6.2.1; 6.2.2.1; 6.2.2.2; 6.2.2.3; 6.2.2.5; 6.2.2.6; 6.2.3.1; 0; 6.2.3.3; 6.2.3.5; 6.2.3.4; 6.3.1; 6.3.2.1; 6.3.2.2; 6.3.3.1; 6.3.3.2; 6.3.5.1; 6.3.5.2.1;

6.3.6.2.1; 6.3.6.3; 6.3.7.2; 6.3.7.3; 6.3.7.4; 6.3.8.1; 6.3.8.2.1; 6.3.8.2.3; 6.3.8.3.1; 6.3.8.3.3; 6.3.9.1; 6.3.9.3.1; 6.3.9.3.2; 6.3.9.4.1; 6.3.9.4.2; 0; 6.3.10.1; 6.3.10.2.1; 6.3.10.2.2; 6.3.10.3; 7.1.1; 7.1.2; 7.2.1; 7.2.2; 7.3.1; 7.3.2.1; 0; 7.3.2.3; 7.5; 8.1; 8.3.1; 8.3.2; 8.3.3.1; 8.3.3.2.1; 8.3.3.2.3; 8.3.3.3.1; 8.3.3.3.2; 8.3.3.3.3; Annex A; Annex B; Annex D.

Mobile app: 3.2.48; 6.2.2.3; 6.2.2.5; 6.3.8.2.3.

Non-interruptible/interruptible cumulative measurement: 3.2.27; 3.2.49; 6.3.2.2; 6.3.6.2.2.

Operating system: 3.2.4; 3.2.51; 6.2.2.5; 0; 6.2.3.7.4; 6.3.5.1; 6.3.5.2.1; 6.3.5.3; 6.3.5.3.1; 6.3.5.3.2; 6.3.5.3.4; 6.3.5.3.5; 6.3.5.3.6; 6.3.8.2.3; 6.3.8.3.1; 6.3.8.3.3; 7.1.2; 7.2.2; 7.3.1; Annex B; Annex D.

Performance: 3.2.15; 7.1.1; 7.2.1; 8.3.3.3.3; Annex A.

Program code: 3.2.52; 3.2.62; 6.3.2.2; 6.3.6.3; 6.3.7.3; 8.2.2.

Protective interface: 3.2.52; 6.2.2.3; 6.2.3.7.1; 6.2.3.7.2; 6.2.3.7.3; 6.2.3.7.4; 6.3.5.1; 6.3.5.3.5; 6.3.8.2.1; 6.3.8.2.3; 6.3.8.3.3; 7.1.2; 7.3.1; Annex B.

Remote verification: 3.2.54; 3.2.68; 6.2.2.1; 6.2.2.7; 6.2.3.3**Error! Reference source not found.**; 6.3.6.1; 6.3.6.2.3; 6.3.7.1; 6.3.10.1; 6.3.10.2.1; 6.3.10.2.2; 6.3.10.2.3; 6.3.10.3; 7.1.2; 7.2.2; 7.3.1; 8.3.1; 8.3.2; 8.3.3.1; 8.3.3.2.1; 8.3.3.2.3; 8.3.3.2.4; 8.3.3.2.5; 8.3.3.3.1; 8.3.3.3.2; 8.3.3.3.3; 8.3.3.3.4; 8.3.3.3.5; Annex B; Annex D.

Seal: 3.2.55; 3.2.64; 5.2; 6.2.3.1; 0; 6.2.3.3**Error! Reference source not found.**; 6.2.3.4; 6.3.2.1; 6.3.5.3.4; 6.3.6.3; 6.3.7.3; 6.3.8.2.3; 6.3.8.3.3; 6.3.9.3.2; 6.3.9.4.2; 7.2.2; 8.2.2; 8.2.3.2; 8.3.1; Annex B; Annex D.

Securing: 3.2.14; 3.2.30; 3.2.56; 4.3; 6.2.2.3; 0; **Error! Reference source not found.**; 6.2.3.4; 6.2.3.5; 6.2.3.6; 6.3.4.2; 6.3.5.3.4; 6.3.6.3; 6.3.7.3; 6.3.8.2.1; 6.3.8.2.3; 6.3.8.3.3; 6.3.9.2; 6.3.9.3.2; 6.3.9.4.2; 0; 6.3.10.3; 7.1.2; 7.2.2; 7.3.1; 8.1; 8.3.1; 8.3.3.3.1; Annex A; Annex B; Annex D.

Software examination: 3.2.60; 5.2; 6.2.2.2; 7.3.2.5; 7.3.2.6; 8.1; Annex D.

Software identification: 3.2.61; 6.2.2.1; 6.2.2.7; 6.2.3.4; 6.3.5.2.1; 6.3.6.1; 6.3.7.1; 6.3.8.3.2; 6.3.8.3.3; 6.3.10.2.2; 7.1.2; 7.2.2; 7.3.1; 7.3.2.3; 8.1; 8.2.2; 8.2.4; 8.3.2; 8.3.3.1; 8.3.3.2.1; 8.3.3.2.5; Annex B; Annex D.

Software interface: 3.2.62; 3.2.65; 7.1.1; 7.3.2.4.

Software module: 3.2.7; 3.2.11; 3.2.12; 3.2.16; 3.2.20; 3.2.31; 3.2.32; 3.2.52; 3.2.58; 3.2.61; 3.2.62; 3.2.63; 3.2.65; 3.2.68; 3.2.73; 3.2.75; 6.2.1; 6.2.2.1; 6.2.2.5; 0; 6.2.3.7.2; 6.2.3.7.4; 6.3.2.1; 6.3.3.1; 6.3.4.1; 6.3.4.2; 6.3.5.2.1; 0; 6.3.5.3.4; 6.3.5.3.5; 6.3.5.3.6; 6.3.6.2.1; 6.3.6.2.2; 6.3.6.3; 6.3.7.2; 6.3.7.3; 6.3.8.1; 6.3.8.2.1; 6.3.8.3.1; 6.3.8.3.3; 6.3.9.4.4; 6.3.10.2.1; 6.3.10.3; 7.1.1; 7.1.2; 7.2.2; 7.3.1; 0; 7.3.2.3; 7.3.2.4; 7.3.2.5; 7.3.2.6; 7.5; 8.1; 8.3.2; Annex B; Annex D.

Software protection: 3.2.55; 3.2.64; 6.2.2.3; 6.2.3.1; 0; 6.3.4.2; 6.3.5.3.4; 6.3.6.3; 6.3.7.3; 6.3.8.2.3; 6.3.9.3.2; 7.3.1; 7.3.2.3; 8.1; Annex B; Annex D.

Software separation: 3.2.65; 6.2.2.1; 6.2.2.5; 6.3.8.2.1; 6.3.8.3.1; 6.3.8.3.3; 7.1.2; 7.3.1; 7.3.2.4; Annex B.

Source code: 3.2.66; 7.1.2; 7.3.1; 0; 7.3.2.4; 7.3.2.5; 7.3.2.6; Annex B.

Storage device: 3.2.67; 6.3.6.2.2; 6.3.6.2.3; 6.3.6.3; Annex B.

Test: 3.2.68; 3.3; 5.1; 6.2.2.2; 6.2.2.6; 6.3.4.1; 6.3.9.4.1;6.3.9.4.4; 6.3.10.2.2; 6.3.10.3; 7.1.1;

7.1.2; 7.2.1; 7.2.2; 7.3.1; 0; 7.3.2.3; 7.3.2.6; 7.4; 7.5; 8.1; 8.3.2; 8.3.3.1; 8.3.3.2.1; 8.3.3.2.3; 8.3.3.2.4; 8.3.3.2.5; 8.3.3.3.1; 8.3.3.3.2; 8.3.3.3.3; 8.3.3.3.4; 8.3.3.3.5; Annex A; Annex B; Annex D.

Timestamp: 3.2.1; 3.2.64; 3.2.69; 6.2.2.6; **Error! Reference source not found.**; 6.2.3.5; 6.3.5.3.1; 6.3.6.2.1; 6.3.7.2; 6.3.8.2.3; 0; 6.3.10.2.1; 7.3.1; Annex B; Annex C.

Transmission of measurement data: 3.2.71; 0; 6.3.7.1; 6.3.7.3; 6.3.7.4; 6.3.8.2.1; 7.3.1; 8.3.2; Annex B; Annex D.

Type-specific parameter: 3.2.31; 3.2.73.

Type evaluation authority: 3.2.50; 6.2.3.1; 6.2.3.7.1; 6.2.3.7.2; 6.3.8.2.1; 6.3.8.3.3; 6.3.10.2.3; 7.1.2; Annex B.

Universal device: 3.2.74; 5.2; 0; 6.3.5.3.6; 6.3.7.3; 6.3.8.2.1; 6.3.8.2.3; 6.3.8.3.3; Annex D.

User interface: 3.2.75; 6.2.2.1; 6.2.2.5; 6.2.3.7.2; 7.1.2; 7.3.1; 7.3.2.3; Annex B.

Verification: 3.2.54; 3.2.68; 3.2.76; 3.2.77; 3.2.78; 3.2.79; 6.2.2.1; 6.2.2.7; **Error! Reference source not found.**; **Error! Reference source not found.**; 6.3.4.2; 6.3.5.2.1; 6.3.6.1; 6.3.6.2.2; 6.3.6.2.3; 6.3.6.3; 6.3.7.1; 6.3.9.1; 6.3.9.3.2; 0; 6.3.10.1; 6.3.10.2.1; 6.3.10.2.2; 6.3.10.2.3; 6.3.10.3; 7.1.1; 7.1.2; 7.2.1; 7.2.2; 7.3.1; 0; 7.3.2.3; 7.3.2.6; 7.4; 8.1; 8.2; 8.2.1; 8.2.3.1; 8.3.1; 8.3.2; 8.3.3.1; 8.3.3.2.1; 8.3.3.2.3; 8.3.3.2.4; 8.3.3.2.5; 8.3.3.3.1; 8.3.3.3.2; 8.3.3.3.3; 8.3.3.3.4; 8.3.3.3.5; Annex B; Annex D.