International Organization of Legal Metrology

Organisation Internationale de Métrologie Légale

Significance of OIML D31 Requirements for Smart Metering

Ulrich Grottker

PTB, Germany

OIML *Seminar on Smart Meters*

Brijuni, Croatia – 2-5 June 2009

Overview

- OIML TC5 / SC2
  - Sub-committee for Software
- D31 Software Requirements
  - General Requirements
  - Specific Requirements
- D31 Validation Methods
  - Simple Methods
  - Advanced Methods
- Summary

TC 5: Electronic measuring instruments and software
MIRS (Slovenia)

TC 5 / SC 1: General requirements for measuring instruments
NMi (The Netherlands)

TC 5 / SC 2: Software
PTB (Germany) + BIML

TC 5/SC 2: Software

P - Members

Australia
Belgium
Belarus
Brazil
Canada
China
Cuba
Czech Republic
Denmark
Finland
France
Germany
Japan
The Netherlands
Norway
Romania
Russia
Slovenia
United Kingdom
U.S.A

O - Members

Austria
Bulgaria
Egypt
Indonesia
Ireland
Mexico
New Zealand
Poland
Slovakia
South Africa
Spain
Sweden
Switzerland
Serbia
+ Liaisons

Basis for the Document

Aims in Legal Metrology according to the Principles of OIML:

*Obtain **confidence in measurements** for trade, surveillance, environment, and safety*

Consequences regarding Software:

- *Correctness of software*
- *Protection of software*
- *Conformity of each instrument with the examined pattern*

D31 Contents

1 Introduction

2 Scope and field of application

3 Terminology

4 Instructions for use of this Document in drafting OIML Recommendations

5 Requirements for measuring instruments with respect to the application of software

- 5.1 General requirements
- 5.2 Requirements specific for configurations

6 Type approval

7 Verification

8 Assessment of severity (risk) levels

Annexes

D31 Contents

1 Introduction

2 Scope and field of application

3 Terminology

4 Instructions for use of this Document in drafting OIML Recommendations

5 Requirements for measuring instruments with respect to the application of software

5.1 General requirements

5.2 Requirements specific for configurations

6 Type approval

7 Verification

8 Assessment of severity (risk) levels

Annexes

Structure of the requirement part

Set of requirements is variable, depending on the features and complexity of the measuring system:

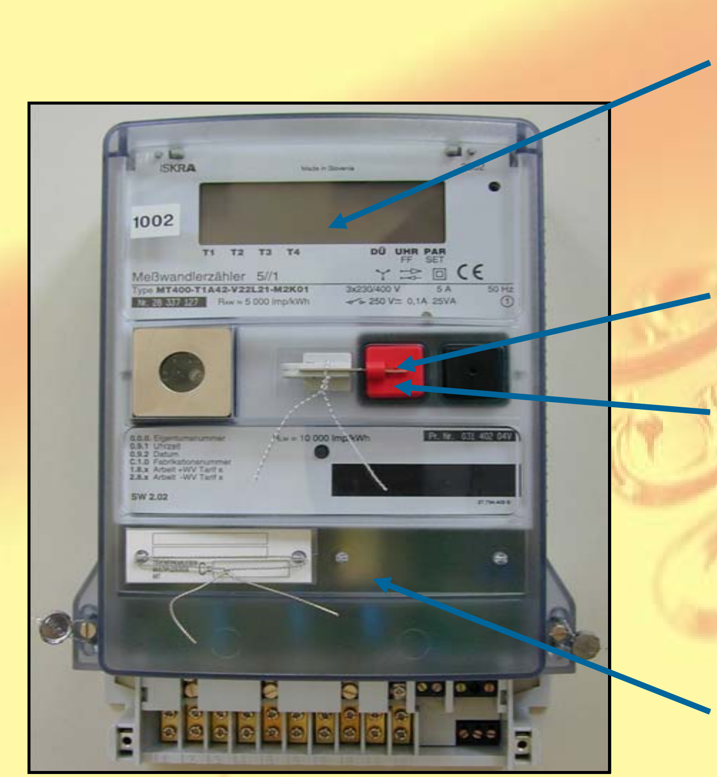
General software requirements (all instruments)
Specific software requirements (specific configurations)

Two severity levels for acceptable technical solutions, depending on the area of application, kind of measurement:

(I) Normal severity level
(II) Raised severity level

5.1 General Requirements > Examples

Example 1: “Simple” instrument (e.g. electricity meter)



5.1.1 Indication of the software identification
5.1.2 Correctness of algorithms and functions, correct units and accompanying information on display

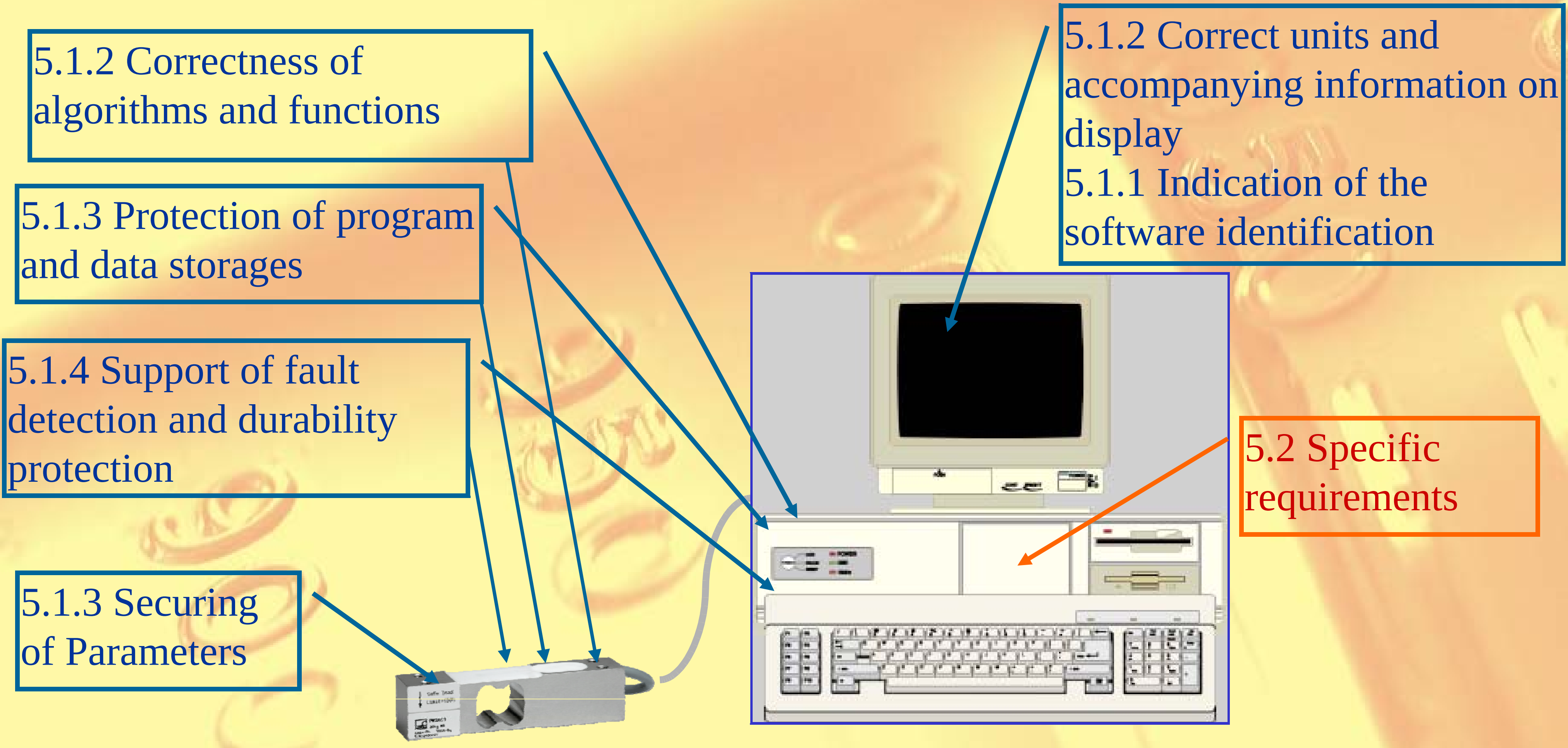
5.1.3 Protection of program and data storages

5.1.3 Securing of Parameters

5.1.4 Support of fault detection and durability protection

5.1 General Requirements > Examples

Example 2: Measuring system (e.g. weighing machine)



5.1.2 Correctness of algorithms and functions

5.1.3 Protection of program and data storages

5.1.4 Support of fault detection and durability protection

5.1.3 Securing of Parameters

5.1.2 Correct units and accompanying information on display

5.1.1 Indication of the software identification

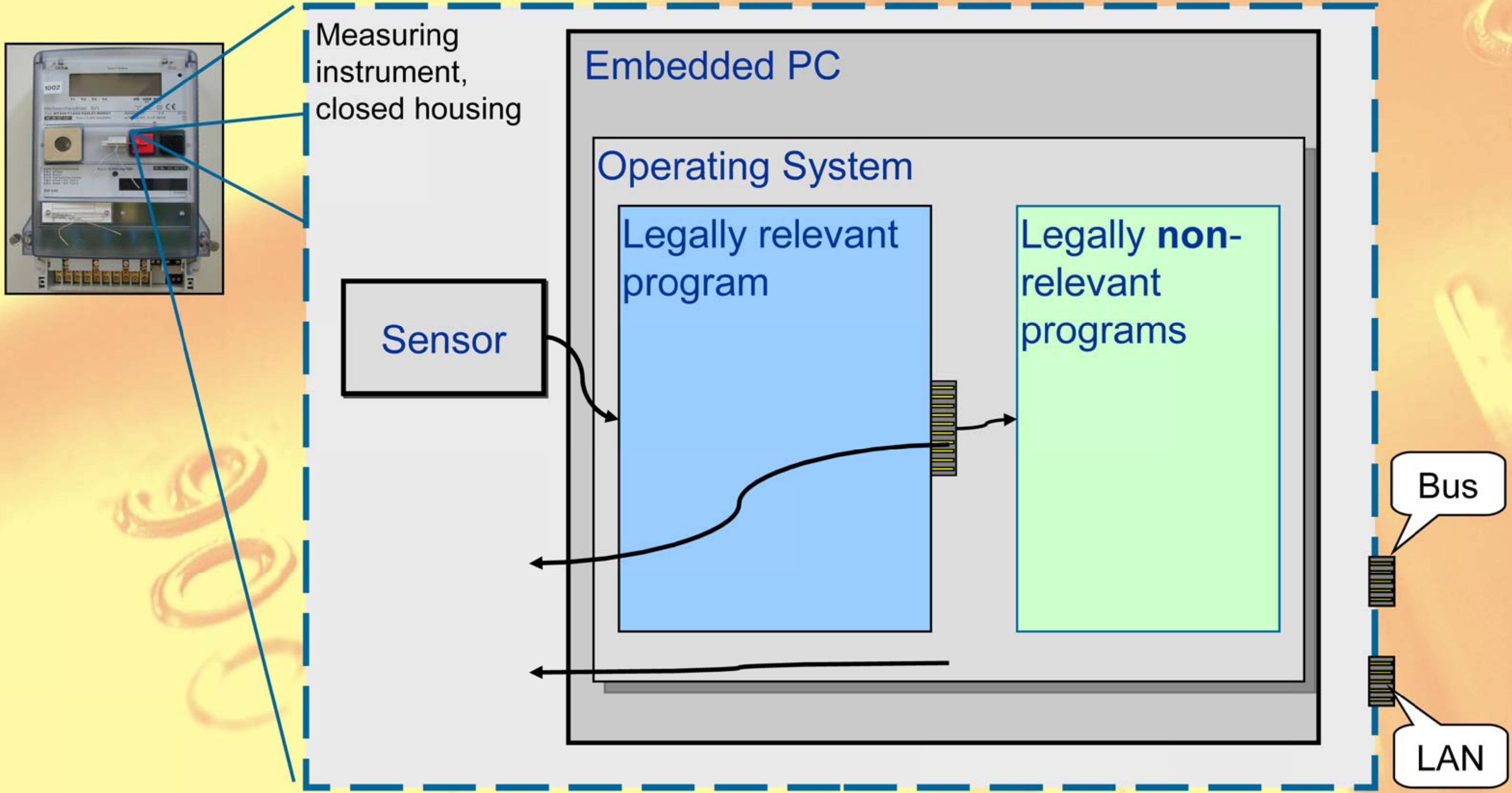
5.2 Specific requirements

Load cell

© Wikipedia/Michael Laible/HBM

5.1 General Requirements > Examples

Example: Future (?) Smart Meter


Measuring instrument, closed housing
Embedded PC
Operating System
Legally relevant program
Legally **non**-relevant programs
Sensor
Bus
LAN

5.1 General requirements > 5.1.1 Software identification

The legally relevant software … shall be clearly identified…

- Identification inextricably linked with the software
- Displayed or printed at start-up, on demand, or continuously

(I) V1.5.12a

V1.5.12a-2D7F (II)

- Output on the display of the instrument or on another sub-assembly

  Exception under certain conditions.

5.1 General requirements > 5.1.2 Correctness of algorithms …

The measuring algorithms and functions … shall be appropriate and functionally correct …

- Take MPE into consideration when constructing or examining an algorithm
- Examples for obviously essential impact of the software to the MPE
  - quantization, number of digits, rounds in successive approximation and abort criterion
  - arithmetic, type of numbers (Integer, float …)
  - analogue-digital conversion
- Specific complex applications
  - image processing
  - dynamic weighing

5.1 General requirements > 5.1.3 Software protection > 5.1.3.1 Prevention of misuse

Protection of program code, measurement values and other relevant data, parameters

… possibilities for … misuse shall be minimal.
Presentation of the measurement result should be unambiguous for all parties affected.

Example: Guided menus for the user for crucial actions
Keep man-machine-interface simple

5.1 General requirements > 5.1.3 Software protection > 5.1.3.2 Fraud protection

a) The legally relevant software shall be secured against unauthorized modification, loading, or changes by swapping the memory device.

Examples:

- The housing containing the memory devices is sealed or the memory device is sealed on the PCB.
- Rewritable device: write-enable input inhibited by a switch that can be sealed. …

(I) (II)

b) Only clearly documented functions are allowed to be activated by the user interface …

Example:

- All inputs from the user interface are redirected to a programme that filters incoming commands. It only allows and lets pass the documented ones and discards all others.

(I) (II)

c) Parameters that fix the legally relevant characteristics of the measuring instrument shall be secured against unauthorized modification. …

Example:

- Device specific parameters to be secured are stored in a non-volatile memory. The write-enable input of the memory is inhibited by a switch that can be sealed.

(I) (II)

d) Software protection comprises appropriate sealing by mechanical, electronic and/or cryptographic means, making an unauthorized intervention impossible or evident.

Example (Electronic sealing):

- The metrological parameters of an instrument can be input and adjusted by a menu item.
- The software recognises each change and increments an event counter.
- The event counter value can be indicated.
- The initial value is registered (e.g. on the plate).
- If the indicated value differs from the registered one, the instrument is in an unverified state (equivalent to a broken seal).

(I)

5.1 General requirements > 5.1.4 Support of hardware features > 5.1.4.1 Support of fault detection

Fault detection described in D11 may be supported by software. … An appropriate reaction on the fault is required.

Example:

- On each start-up the legally relevant program calculates a checksum of the program code and legally relevant parameters.
- The nominal value of these checksums has been calculated in advance and stored in the instrument.
- If the calculated and stored values don't match, the program stops execution.

(I) (II)

5.1 General requirements > 5.1.4 Support of hardware features > 5.1.4.2 Support of durability protection

Durability protection described in D11 may be supported by software. … An appropriate reaction on the fault is required.

Example:

- Some kinds of measuring instruments need an adjustment after a prescribed time interval.
- The software gives a warning when the maintenance interval has elapsed.
- The software stops measuring, if it has been exceeded for a certain time interval.

(I) (II)

D31 Contents

1 Introduction

2 Scope and field of application

3 Terminology

4 Instructions for use of this Document in drafting OIML Recommendations

5 Requirements for measuring instruments with respect to the application of software

- 5.1 General requirements
- 5.2 Requirements specific for configurations

6 Type approval

7 Verification

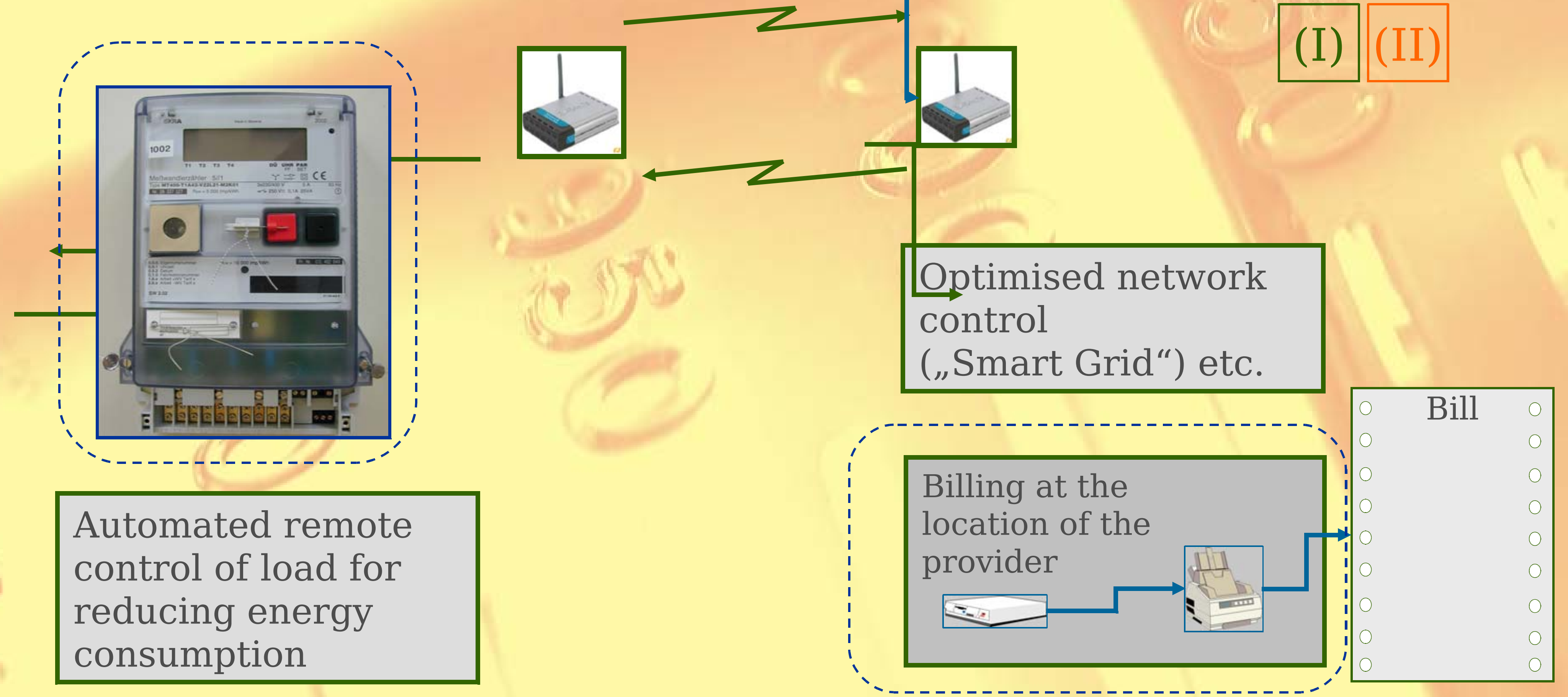
8 Assessment of severity (risk) levels

Annexes

5.2 Specific requirements > 5.2.1 Parts, interfaces of parts > 5.2.1.1 Electronic devices and sub-assemblies

a) Sub-assemblies or electronic devices of a measuring system that perform legally relevant functions shall be identified, clearly defined, and documented. They form the legally relevant part of the measuring system.

Example: Smart meter



Automated remote control of load for reducing energy consumption

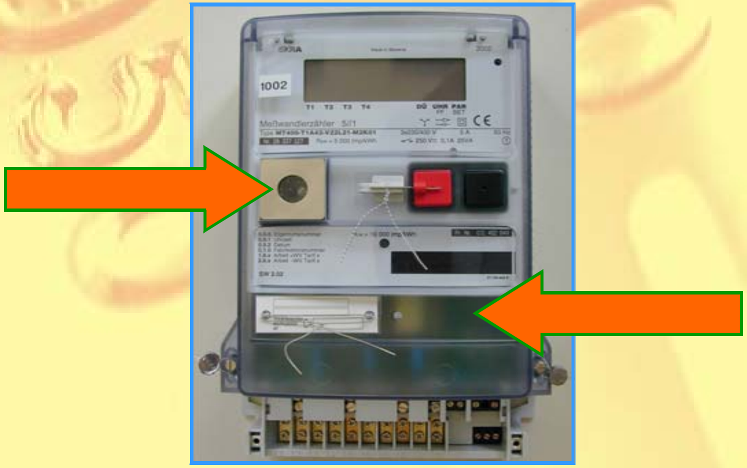
Optimised network control („Smart Grid“) etc.

Billing at the location of the provider

5.2 Specific requirements > 5.2.1 Parts, interfaces of parts > 5.2.1.1 Electronic devices and sub-assemblies

b) Relevant functions and data of sub-assemblies and electronic devices shall not be inadmissibly influenced by commands received via the interface.

This implies that there is an unambiguous assignment of each command to all initiated function or data change in the sub-assembly or electronic device.



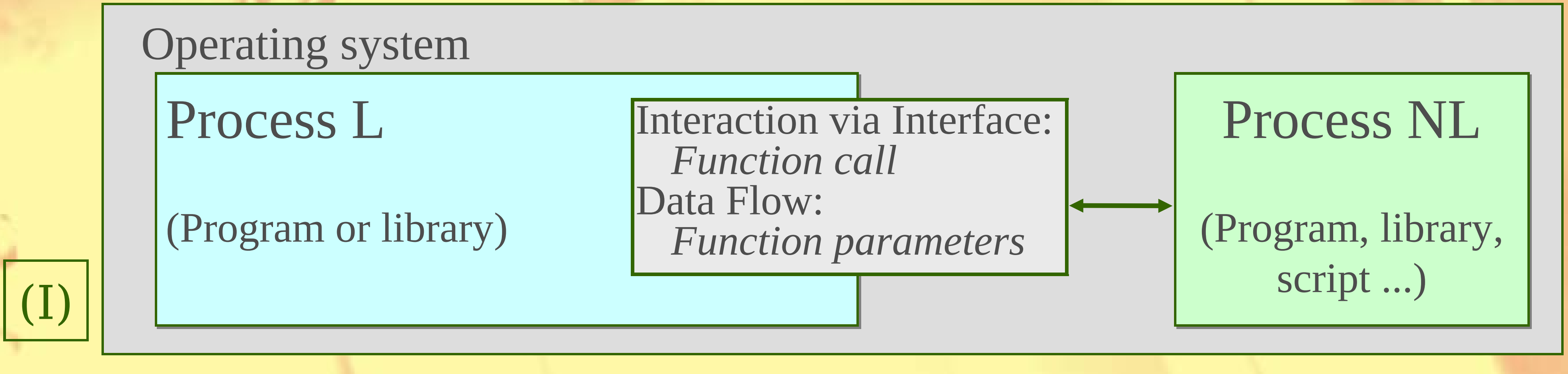
5.2 Specific requirements > 5.2.1 Parts, interfaces of parts > 5.2.1.2 Separation of software parts

a) All software modules (programs, subroutines, objects etc.)
that perform legally relevant ***functions*** or
that contain legally relevant ***data domains***
form the legally relevant software part of a measuring instrument.

b) All communication shall be performed ***exclusively via*** the software ***interface***.

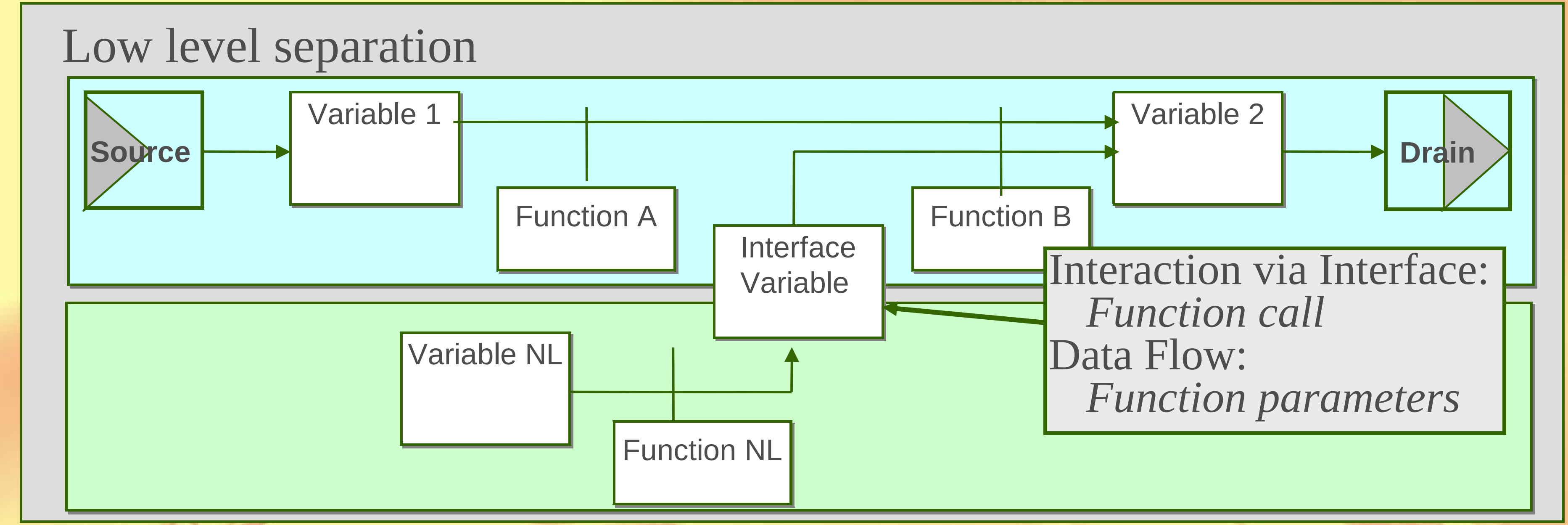
c) Same requirements as for hardware interfaces concerning ***commands and data flow***.

Example (1): High level separation, using features of the operating system



5.2 Specific requirements > 5.2.1 Parts, interfaces of parts > 5.2.1.2 Separation of software parts

Example (2): Low level separation, using features of the programming language



Interaction via Interface:
Function call
Data Flow:
Function parameters

(I)

d) The legally relevant software shall have ***priority*** using the resources over non-relevant software.
The measurement task … must not be delayed or blocked by other tasks.

5.2 Specific requirements > 5.2.2 Shared indications

a) … The window containing the legally relevant data shall have ***highest priority*** i.e.

- it shall not be deleted by other software
- or overlapped by windows generated by other software
- or minimised
- or made invisible

as long as the measurement is running and the presented results are needed for the legally relevant purpose.

5.2 Specific requirements > 5.2.3 Storage, transmission of data > 5.2.3.1 … 5.2.3.7

5.2.3.1 The measurement value stored or transmitted shall be accompanied by ***all relevant information*** necessary for future legally relevant use

5.2.3.2/3 The data shall be ***protected*** by software means to guarantee authenticity, integrity / correctness of the information of the time of measurement. … If an irregularity is detected, the data shall be discarded or marked unusable.

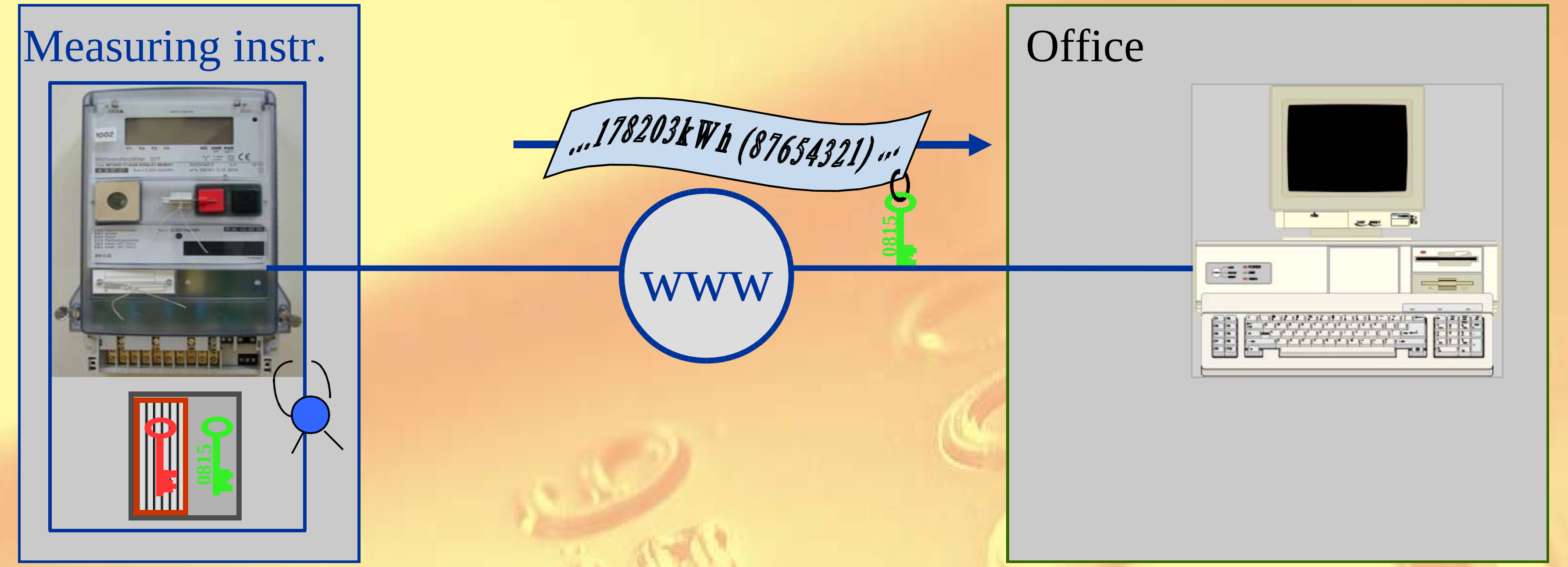
5.2.3.4 Measurement data must be stored ***automatically*** when the measurement is concluded, i.e. when the final value used for the legal purpose has been generated. …

5.2.3.5/6 The measurement shall not be ***inadmissibly influenced*** by a transmission delay. If network services become unavailable, no measurement data shall be lost. …

5.2.3.7 The ***time stamp*** shall be read from the clock of the device. … setting the clock may be legally relevant and appropriate protection means shall be taken …

5.2 Specific requirements > 5.2.3 Storage, transmission of data

> 5.2.3.3 Authenticity and integrity of data sets



Hash (**178203 kWh**) = Σ
Σ ⊗ 🔑 = **87654321**

→

Hash (**178203 kWh**) = Σ
87654321 ⊗ 🔑 = Σ

(II)

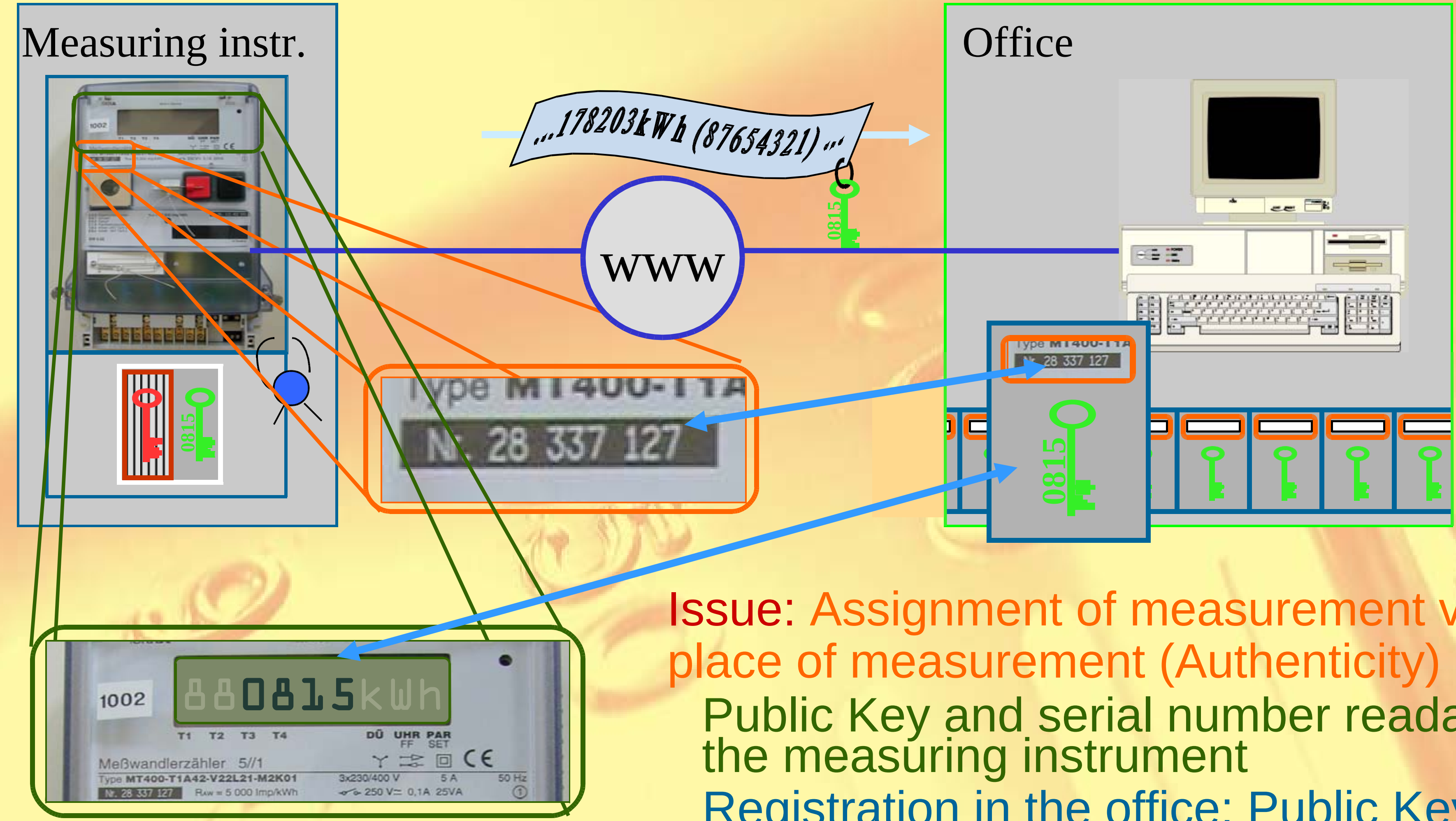
Public Key System: Secret Key hidden in the measuring instrument

Approved algorithms for high security: RSA or „Elliptic Curves“

Relevant data + signature + public key sent: **Integrity** verifiable

5.2 Specific requirements > 5.2.3 Storage, transmission of data

> 5.2.3.3 Authenticity and integrity of data sets



Issue: Assignment of measurement value – place of measurement (Authenticity)

Public Key and serial number readable at the measuring instrument

Registration in the office: Public Key + Serial no.+ place of measurement

(II)

→ **Authenticity** verifiable

5.2 Specific requirements > 5.2.4 Compatibility, portability

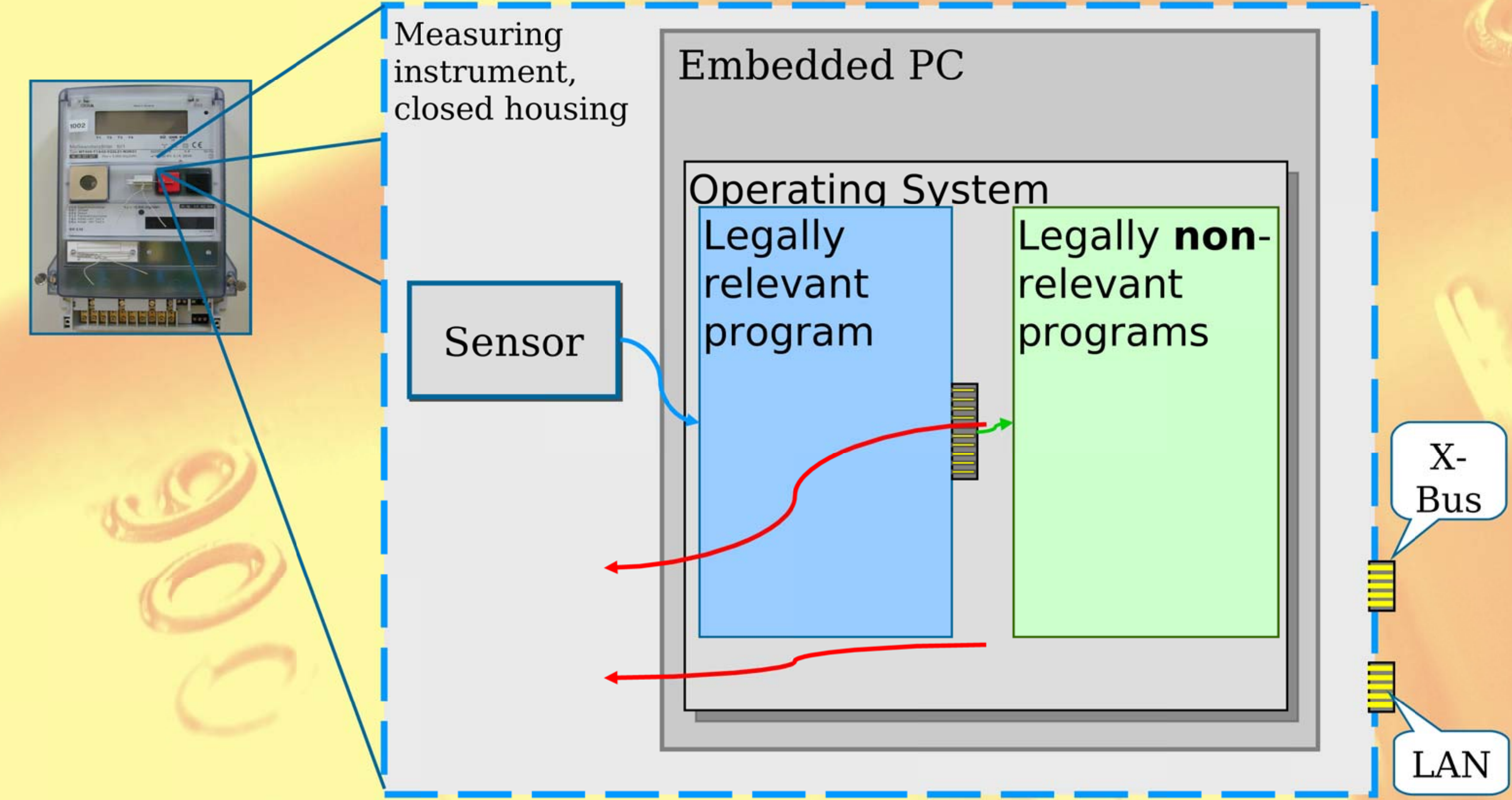
5.2.4.1 The manufacturer shall identify the ***hardware and software environment***, minimal resources and a suitable configuration …

5.2.4.2 Technical means shall be provided in the legally relevant software to ***prevent operation***, if the minimal configuration requirements are not met.

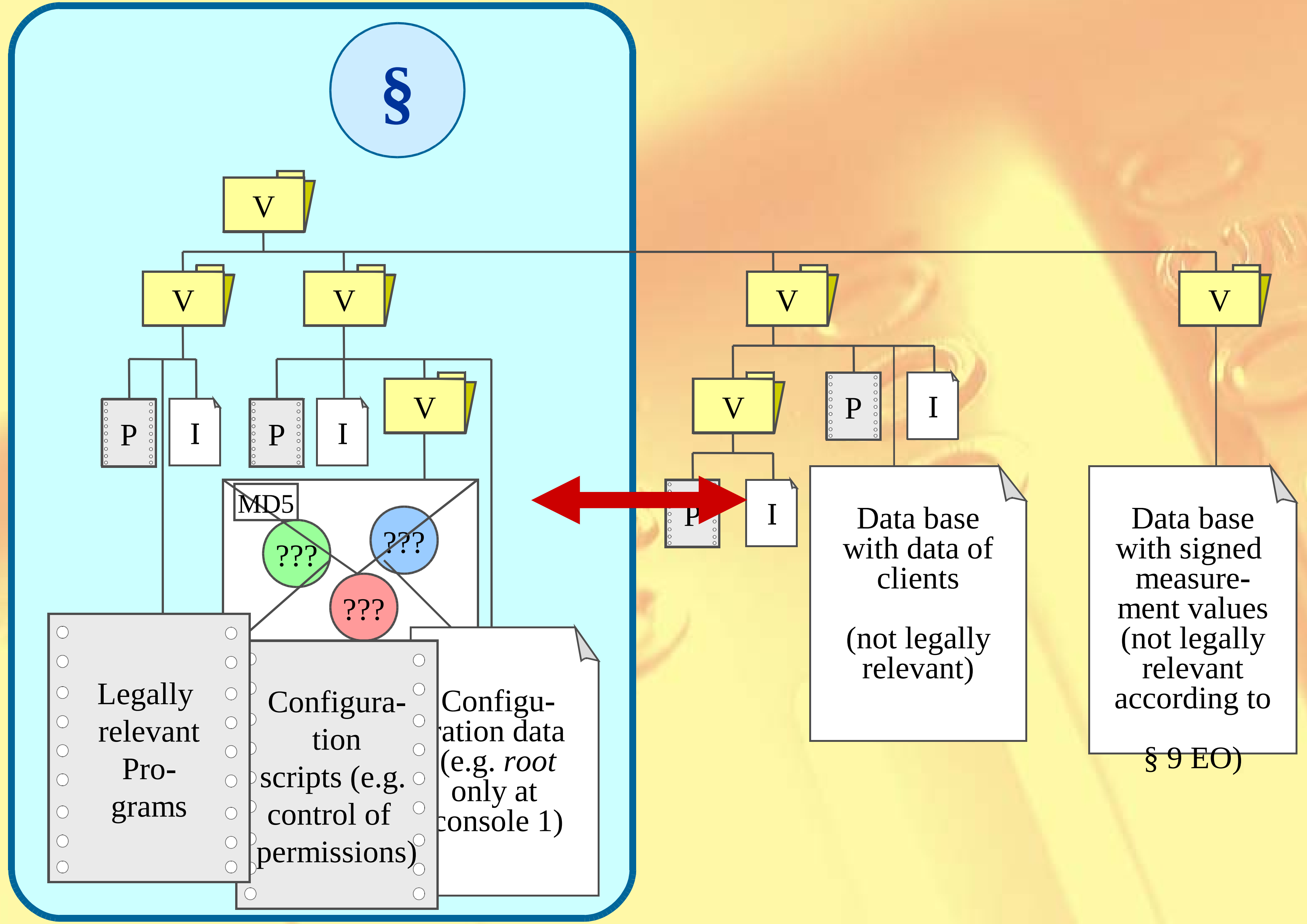
The system shall be ***operated only*** in the ***environment specified*** by the manufacturer for its correct functioning.

… in case an invariant environment is specified …, means shall be provided to keep the ***operating environment fixed***.

Example: Future (?) Smart Meter



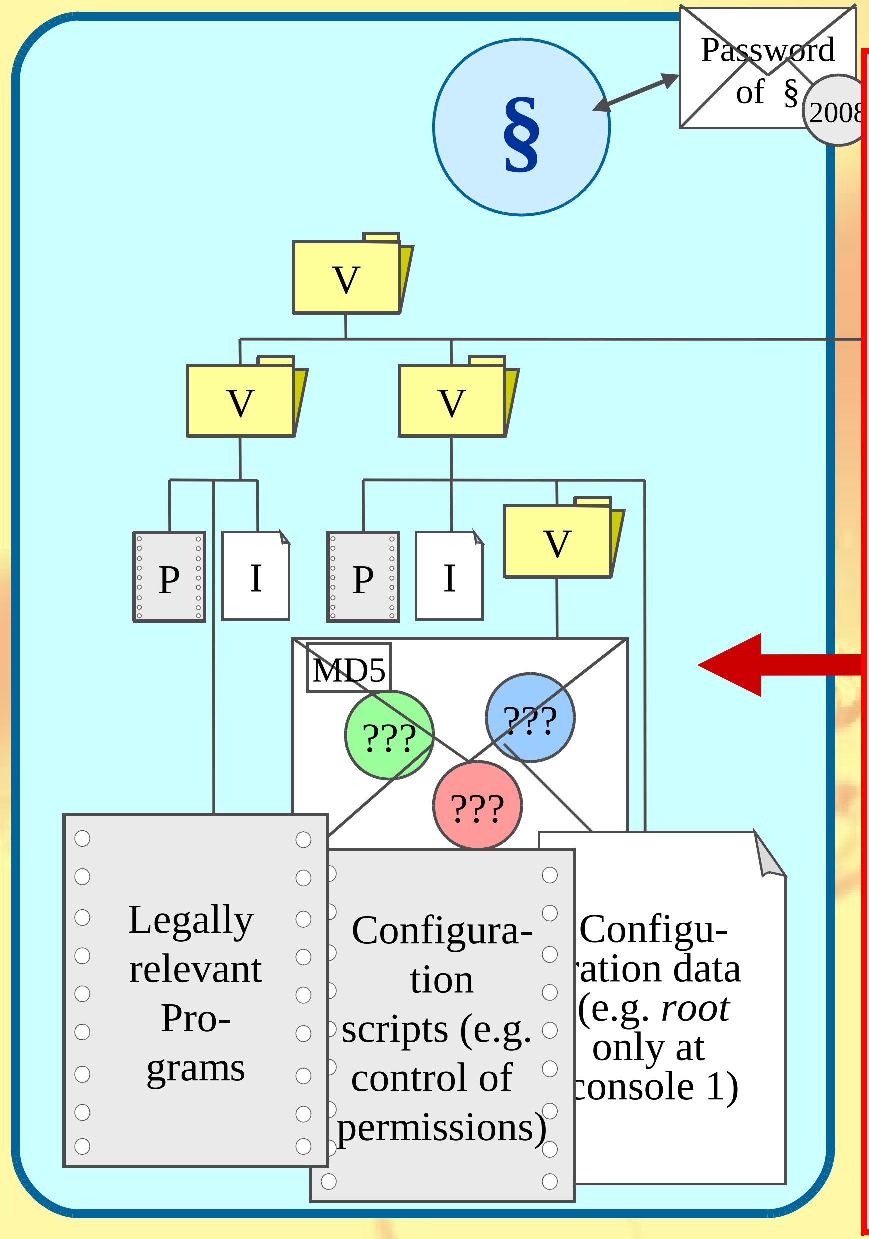
5.2 Specific requirements > 5.2.4 Compatibility, portability


§
V
V
V
V
V
P
I
P
I
V
V
P
I
P
I
MD5
???
???
???
Legally relevant Pro-grams
Configura-tion scripts (e.g. control of permissions)
Configu-ration data (e.g. *root* only at console 1)
Data base with data of clients

(not legally relevant)
Data base with signed measure-ment values (not legally relevant according to

§ 9 EO)

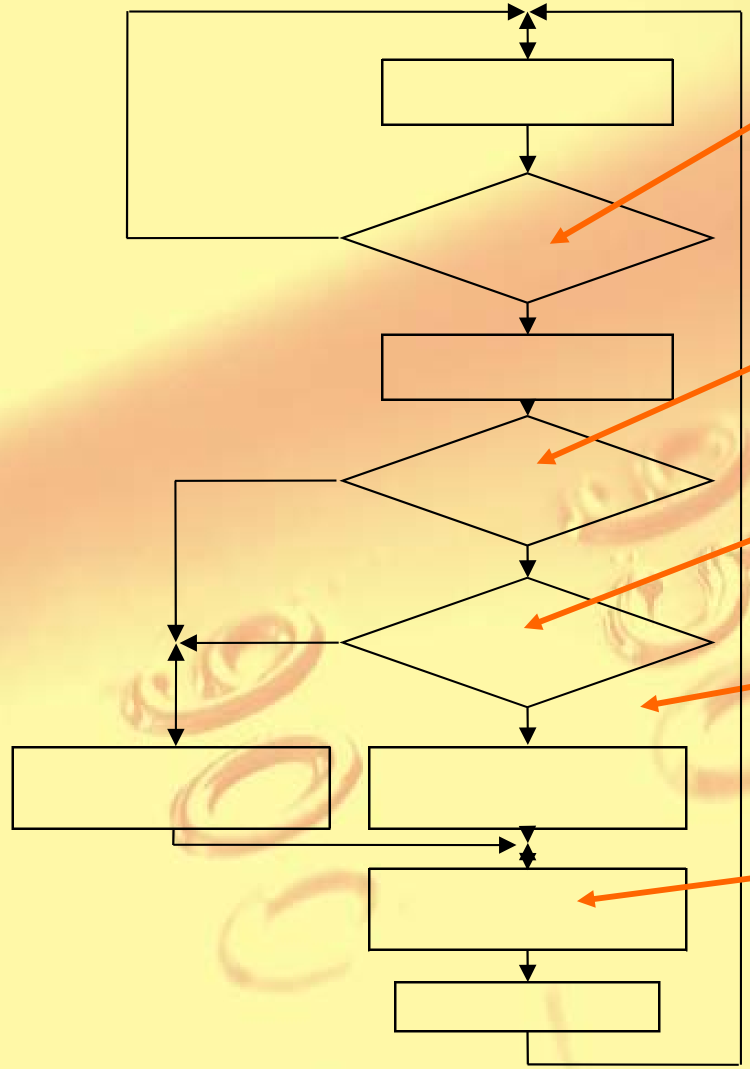
5.2 Specific requirements > 5.2.4 Compatibility, portability


§
Password of § 2008
V
V
V
P
I
P
I
V
MD5
???
???
???
Legally relevant Pro-grams
Configura-tion scripts (e.g. control of permissions)
Configu-ration data (e.g. *root* only at console 1)

Means to keep environment fixed:

- Separate software
- Make use of the multi-tasking protection means of the operating system: lock the role “admin” or “root”.
- Adjust read-write-execute permissions
- Reduce functionalities of the OS by installing “Security Policies”
- Reduce functionality: no plug&play interfaces allowed (USB, Firewire, PCMCIA, …)
- Highest protection of LAN interface (Firewall)

Example for Traced Update procedure



- Traced update automatic.
- Control of traced update by fixed software
- Start only with consent of the user

- Guarantee that integrity is OK

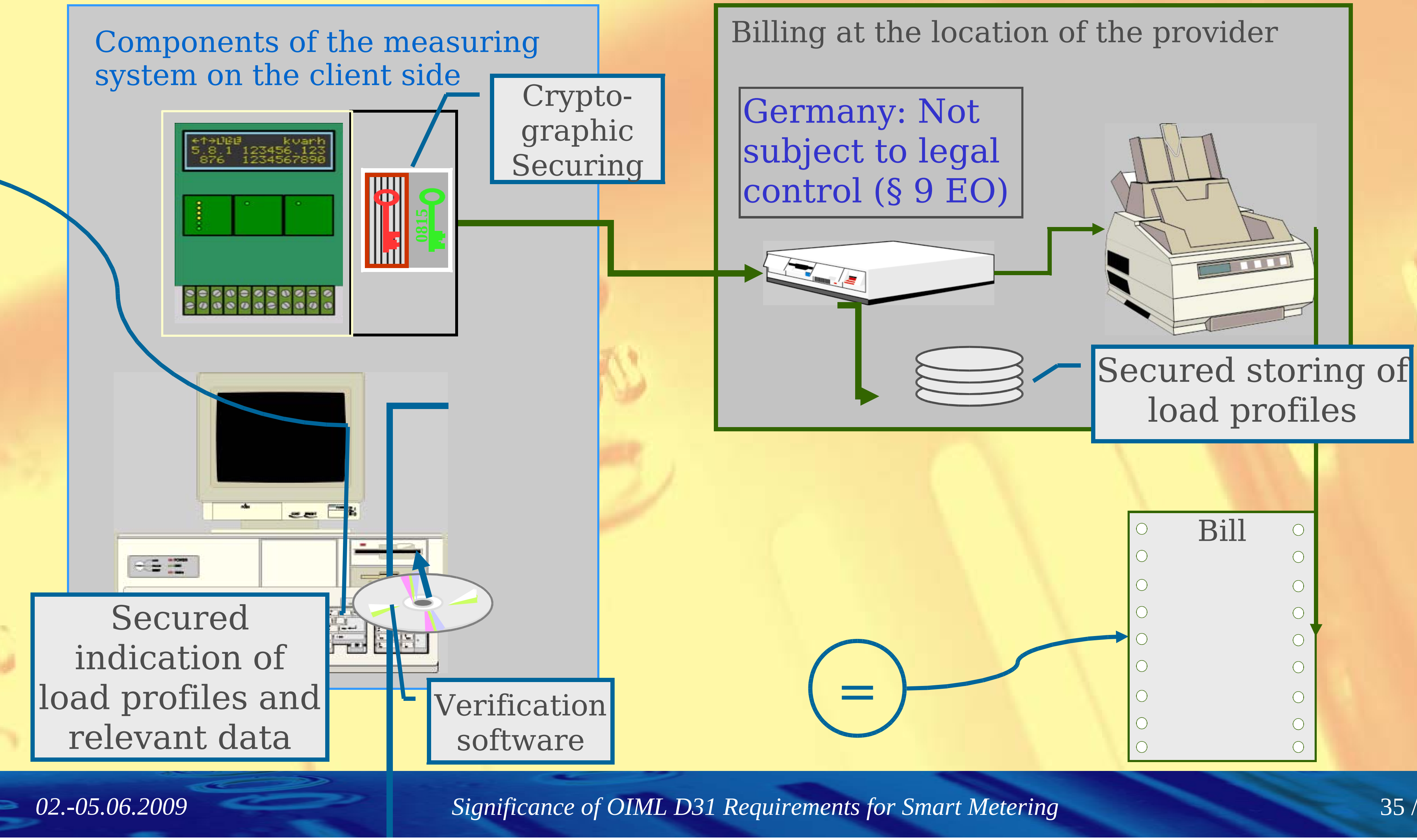
- Guarantee that authenticity is OK (origin known and correct)

- Installation automatic. Fall back to old version if installation fails.

- Audit trail of all update activities
- All relevant steps traceable

Example for a complex Smart Meter (1)

Measuring the load profile locally and storing it centrally



Components of the measuring system on the client side

Crypto-graphic Securing

Secured indication of load profiles and relevant data

Verification software

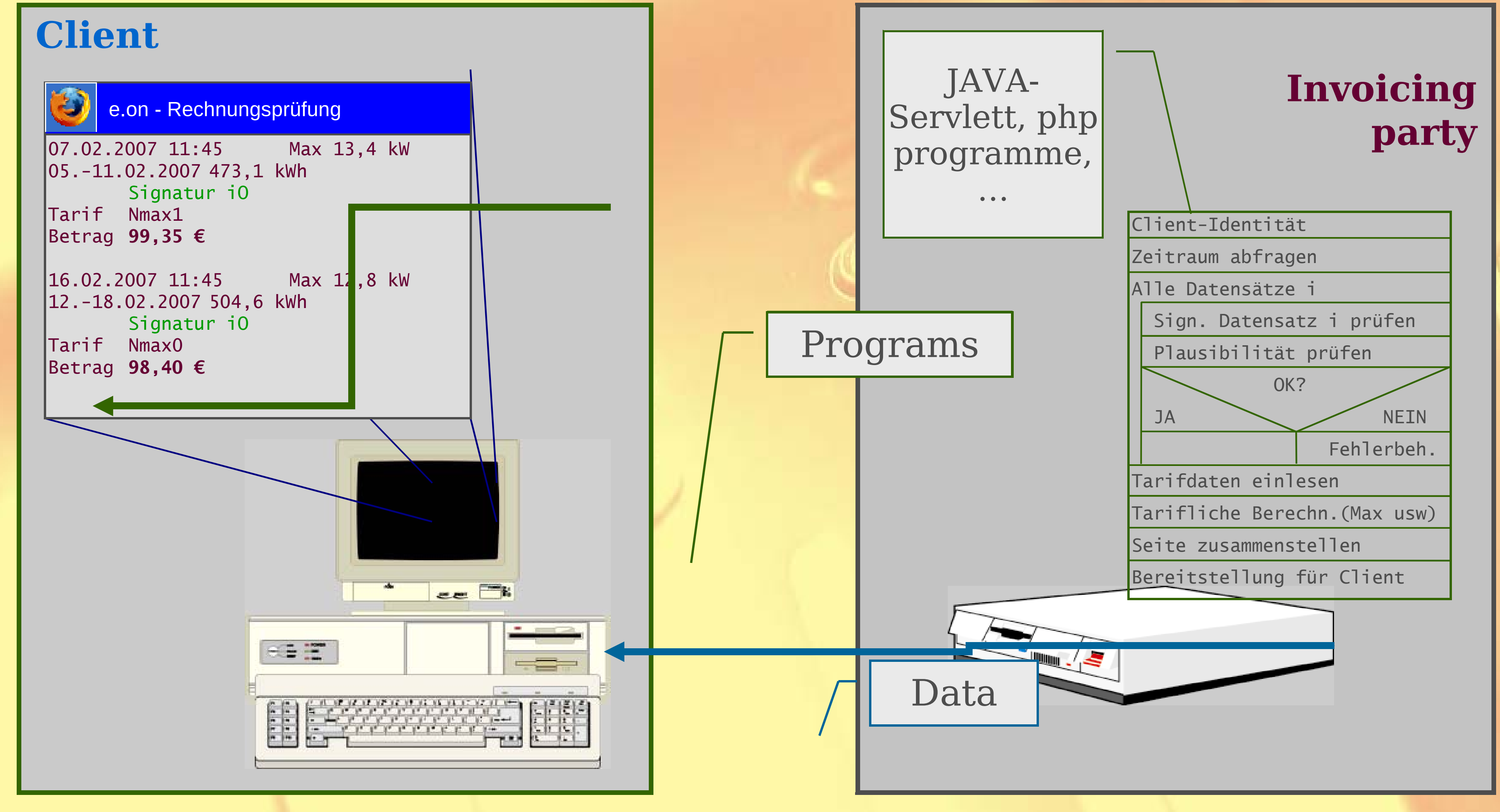
Billing at the location of the provider

Germany: Not subject to legal control (§ 9 EO)

Secured storing of load profiles

Example for a complex Smart Meter (2)

Software for the verification of the bill by the client:
(a) Program stored on the web site of the invoicing party.



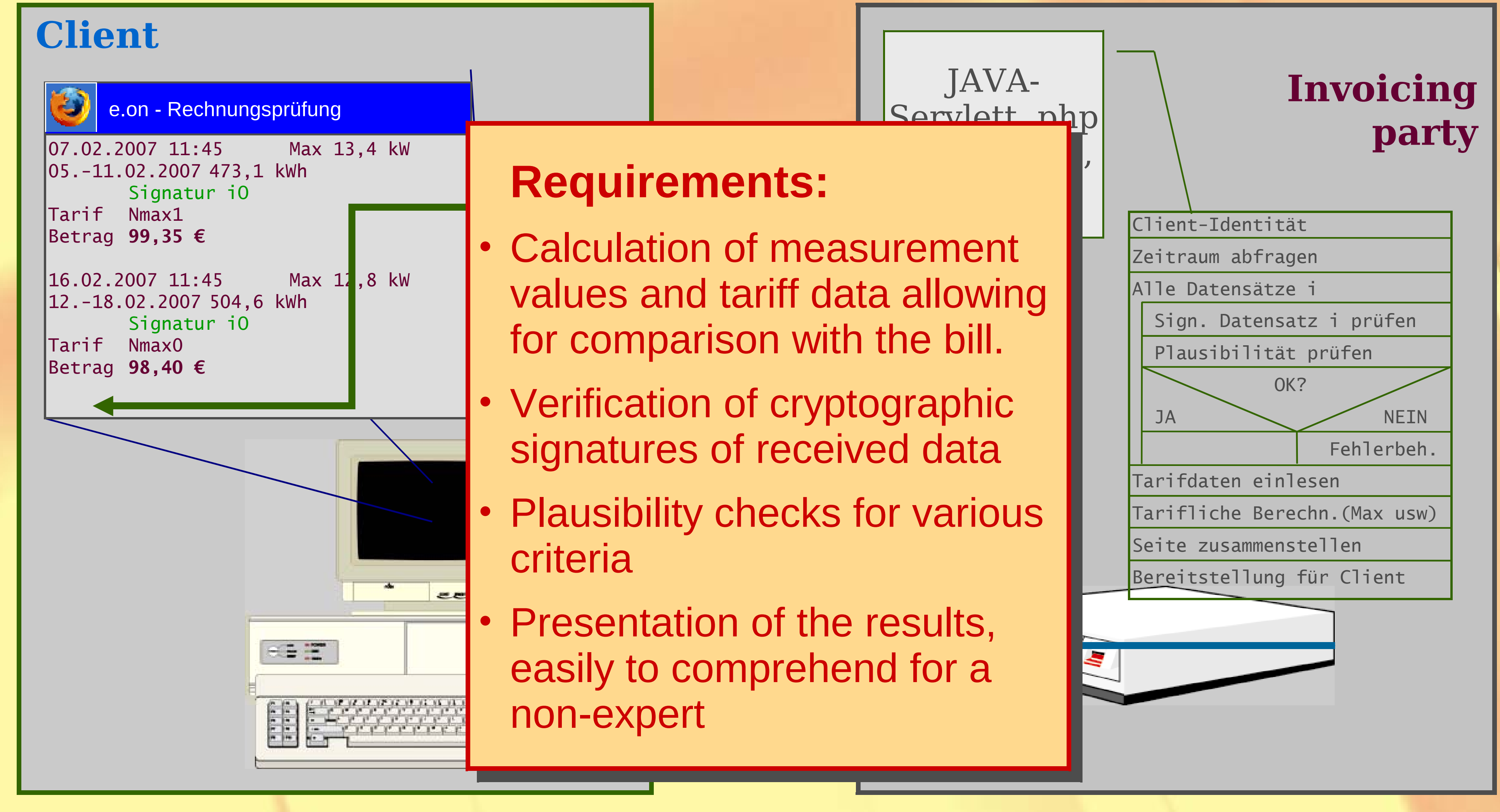
```
07.02.2007 11:45       Max 13,4 kW
05.-11.02.2007 473,1 kWh
              Signatur iO
Tarif   Nmax1
Betrag  99,35 €

16.02.2007 11:45       Max 12,8 kW
12.-18.02.2007 504,6 kWh
              Signatur iO
Tarif   Nmax0
Betrag  98,40 €
```

```
Client-Identität
Zeitraum abfragen
Alle Datensätze i
    Sign. Datensatz i prüfen
    Plausibilität prüfen
    OK?
    JA                  NEIN
                        Fehlerbeh.
Tarifdaten einlesen
Tarifliche Berechn.(Max usw)
Seite zusammenstellen
Bereitstellung für Client
```

Example for a complex Smart Meter (3)

Software for the verification of the bill by the client:
(a) Program stored on the web site of the invoicing party.



Client

e.on - Rechnungsprüfung

```
07.02.2007 11:45       Max 13,4 kW
05.-11.02.2007 473,1 kWh
       Signatur iO
Tarif  Nmax1
Betrag 99,35 €

16.02.2007 11:45       Max 12,8 kW
12.-18.02.2007 504,6 kWh
       Signatur iO
Tarif  Nmax0
Betrag 98,40 €
```

Requirements:

- Calculation of measurement values and tariff data allowing for comparison with the bill.
- Verification of cryptographic signatures of received data
- Plausibility checks for various criteria
- Presentation of the results, easily to comprehend for a non-expert

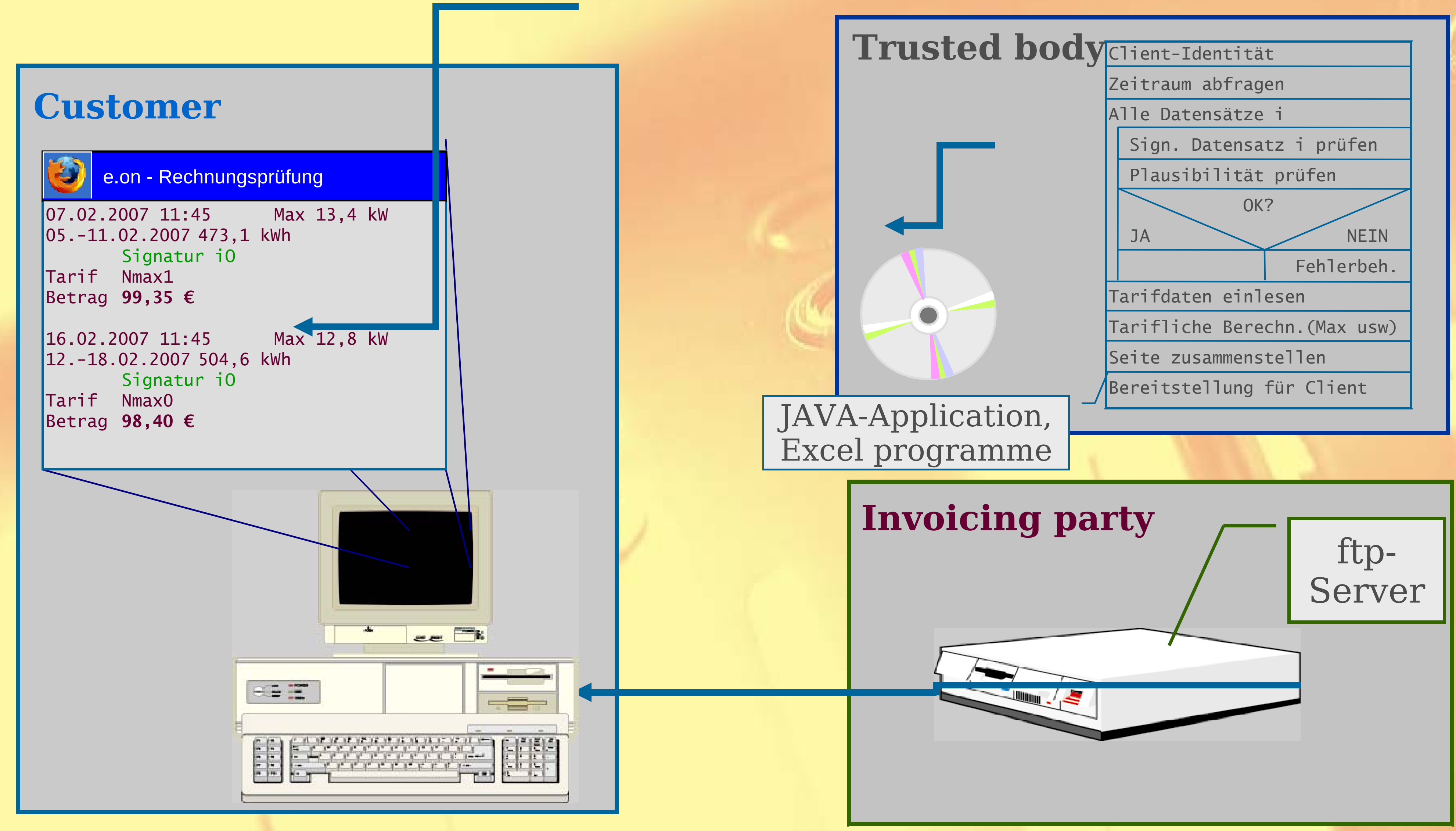
JAVA-
Servlett, php
,

Invoicing party

```
Client-Identität
Zeitraum abfragen
Alle Datensätze i
  Sign. Datensatz i prüfen
  Plausibilität prüfen
              OK?
  JA                    NEIN
                   Fehlerbeh.
Tarifdaten einlesen
Tarifliche Berechn.(Max usw)
Seite zusammenstellen
Bereitstellung für Client
```

Example for a complex Smart Meter (4)

Software for the verification of the bill by the client:
(b) Certified program provided by a trusted body



```
07.02.2007 11:45       Max 13,4 kW
05.-11.02.2007 473,1 kWh
       Signatur iO
Tarif  Nmax1
Betrag 99,35 €

16.02.2007 11:45       Max 12,8 kW
12.-18.02.2007 504,6 kWh
       Signatur iO
Tarif  Nmax0
Betrag 98,40 €
```

```
Client-Identität
Zeitraum abfragen
Alle Datensätze i
  Sign. Datensatz i prüfen
  Plausibilität prüfen
              OK?
  JA                    NEIN
                    Fehlerbeh.
Tarifdaten einlesen
Tarifliche Berechn.(Max usw)
Seite zusammenstellen
Bereitstellung für Client
```

D31 Contents

1 Introduction

2 Scope and field of application

3 Terminology

4 Instructions for use of this Document in drafting OIML Recommendations

5 Requirements for measuring instruments with respect to the application of software

6 Type approval

6.1 Documentation to be supplied for type approval

6.2 Requirements on the approval procedure

6.3 Validation methods

6.4 Validation procedure

6.5 Equipment under test

7 Verification

8 Assessment of severity (risk) levels

Validation methods

Abbr.	Description	Application	Preconditions	Special skills
AD	Analysis of documentation	Always	Documentation	-
VFTM	Validation by functional testing of metrological features	Correctness of the algorithms, uncertainty, compensating and correcting algorithms	Documentation	-
VFTSw	Validation by functional testing of software features	Handling by the user, correct functioning of communication, indication, fraud protection	Documentation text editor	-
DFA	Data flow analysis	Software separation, evaluation of the impact of commands on the instrument’s functions	Source code, text editor	Programming languages.
CIWT	Code inspection, Walkthrough	All purposes	Source code, text editor	Programming languages
SMT	Software module testing	All purposes when input and output can clearly be defined	Source code, testing environment, special software tools	Programming languages. Instruction for using the tools

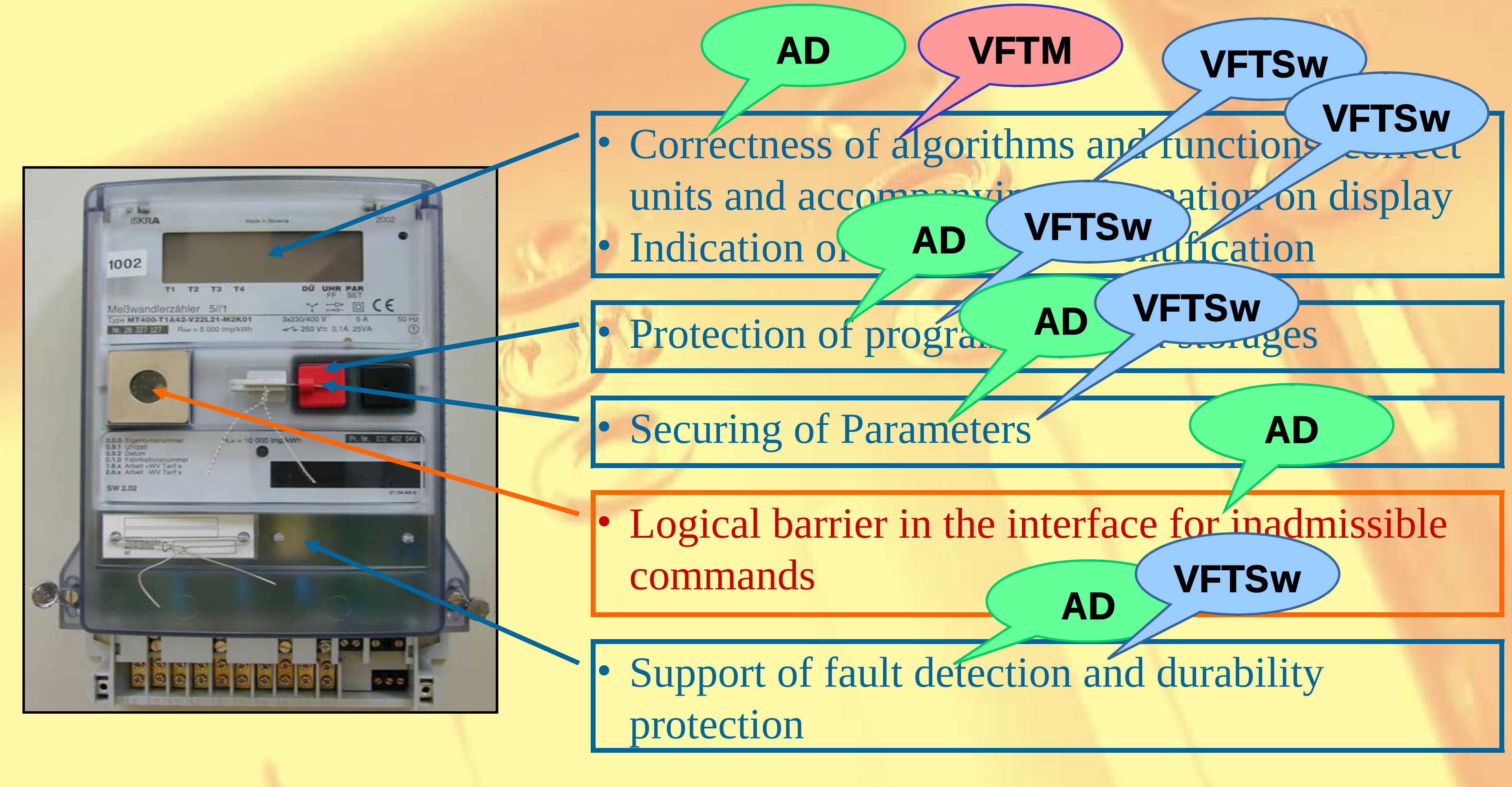
Validation Procedures

	Requirement	Validation procedure A (normal examination level)	Validation procedure B (extended examination level)	Comment
5.1.1	Software identification	AD + VFTSw	AD + VFTSw + WT	Select »B« if high conformity is required
5.1.2	Correctness of algorithms and functions	AD + VFTM	AD + VFTM + WT/SMT	
5.1.3.1	Prevention of accidental misuse	AD + VFTSw	AD + VFTSw	
5.1.3.2	Fraud protection	AD + VFTSw	AD + VFTSw + DFA/WT/SMT	Select »B« in case of high risk of fraud
5.1.4.1	Support of fault detection	AD + VFTSw	AD + VFTSw + WT + SMT	Select »B« if high reliability is required
5.1.4.2	Support of durability protection	AD + VFTSw	AD + VFTSw + WT + SMT	Select »B« if high reliability is required

Validation procedure

Normal examination level

Example: “Simple” instrument (e.g. electricity meter)



- Correctness of algorithms and functions, correct units and accompanyi[illegible]ation on display
- Indication o[illegible]ntification
- Protection of progra[illegible]storages
- Securing of Parameters
- Logical barrier in the interface for inadmissible commands
- Support of fault detection and durability protection

Summary

- **D31 is a basis for future Recommendations**
  - Software requirements based on the Principles of OIML for Legal Metrology
  - Requirements take into consideration current state of Information Technology
- **Smart Meters: Various configurations possible conform to D31**
  - Tools for identifying legally relevant parts of a measuring system
  - New concepts will not be limited more than necessary

Thank you for your attention!