SmartCom

Report on the validation of a demonstrator for the exchange of dimensional measurements in an end user application, with a secure logistic data chain including DCCs

EN

Ceracrane

DOI: 10.5281/zenodo.5522855

DOI 10.5281/zenodo.5522855

Report on the validation of a demonstrator for the exchange of dimensional measurements in an end user application, with a secure logistic data chain including DCCs

Version 1.0

DOI 10.5281/zenodo.5522855

Editors

Aalto University, Finland:
T. Mustapää, L. Immonen, H. Tunkkari, J. Taponen, L. Parkkinen, J. Pousi

Physikalisch Technische Bundesanstalt, Germany:
D. Hutzschenreuter, W. Heeren, C. Brown, O. Baer

Zeiss
R. Bernhardt

Hexagon
A. Hergenräder, C. Herrman

Mitutoyo
K. Stein

Sartorius Lab Instruments
J. Haller

Mettler-Toledo
C. Müller-Schöll

Comprising the results from our research and the fruitful and intensive discussions with all our other project partners worldwide.

Contact: smartcom@ptb.de

Espoo September 2021

DOI 10.5281/zenodo.5522855

Table of Contents

DOI 10.5281/zenodo.5522855

1 Introduction

To test and validate the research outcomes of SmartCom [1] in industrial end-user applications, two demonstrators were developed as a part of the project. In this deliverable report we introduce a demonstrator for exchanging metrological data in a smart overhead crane. The demonstrator showcases the use of digital calibration certificates (DCC) [2], [3], digital SI (D-SI) [4] and appropriate cryptographical methods such as digital signatures and distributed ledger technology (DLT), a.k.a. a blockchain, for secure exchange of measurement data and relevant metadata of cargo containers.

The report is organised as follows: Section 2 provides the relevant background of the use case. The demonstrator concept and implemented functionalities are described in Section 3. The main features of the user interface (UI) and the use of the demonstrator system are discussed in Section 4. Section 5 summarises this report.

DOI 10.5281/zenodo.5522855

2 Background

Millions of tons of goods are being transported all around the world constantly. A major part of that transportation happens through harbours where the goods are typically handled in containers. Since July 2016 the International Convention for the Safety of Life at Sea (SOLAS) has required that the Verified Gross Mass of the containers must be delivered to the vessel carrier as it is needed for the stowage plan of the vessel to optimize the ship's stability [5].

Another important reason for collecting data from the containers is tracing of the cargo. Availability of the weight information makes it easier to notice any significant changes in the containers weight along their route, which helps investigating and preventing smuggling and other kinds of crimes.

DOI 10.5281/zenodo.5522855

3 Concept and implementation

The demonstrator, Ceracrane, was developed at the Aalto University's Industrial Internet Campus (AIIC) using the campus' smart overhead crane to replicate a harbour environment. More information about AIIC and the smart crane can be found from the AIIC website [6] and [7]. The source code of the demonstrator is available at the Ceracrane GitLab repository [8]. Figure 1 presents the architecture of the demonstrator implementation. The functions of the different application programming interfaces (API) are presented in the following subsections.

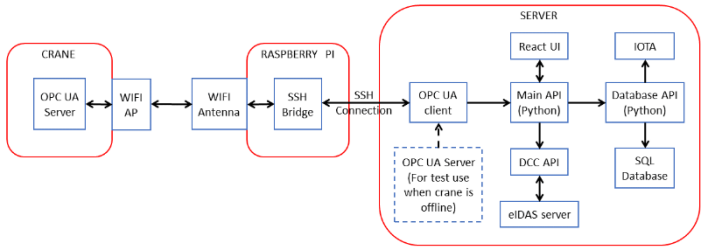

Figure 1. Demonstrator architecture.

3.1 Main API

Main API is used for running the system and managing the measurement creation and search processes. When the measurement data and metadata are obtained through the OPC unified architecture (OPC UA) client, the main API

DOI 10.5281/zenodo.5522855

creates an extensible markup language (XML) file containing the data. The XML format consists of three parts:

- Metadata: this part contains the name or an identifier of the crane that was used for the container handling and obtaining the measurement results, a time stamp and an identification specified for the container.
- Measurement data: this part contains the measurement values collected from the crane's sensors, which are gross and tare weight of the container and the positions of the crane's hoist, trolley, and bridge. The measurement data structured with the D-SI schema making it possible to validate the data with the TraCIM service presented in the SmartCom deliverable D5 [9].
- Signature: the measurement file is signed to prove the origin, authenticity, and integrity of the data when the data is needed.

An example of the XML format excluding the signature is presented in Annex 5.1.

3.2 OPC UA client

OPC UA client is used for obtaining the measurement data and metadata from the crane's OPC UA server. The data exchange between the OPC UA client and OPC UA server was implemented using a Raspberry PI to provide an secure shell (SSH) bridge for ensuring security in an open network environment.

DOI 10.5281/zenodo.5522855

3.3 DCC API

DCC API is used for viewing the DCCs and sending files for the eIDAS server. The DCCs used in the demonstrator are using the DCC schema version 3.0.0-rc2 []. More detailed information about the DCC structure and schema can be found from SmartCom deliverable D3 and []. The DCCs are used in the demonstrator to indicate to properties of the measurement devices in detail and prove that the devices have been maintained correctly, thus ensuring that the measurement data is trustworthy in terms of its quality.

3.4 eIDAS server

EIDAS is a regulation of the European Parliament and Counsil on electronic identification and trust services [10]. An eIDAS signing service is used in the demonstrator for creating digital signatures for the individual measurement files and validating them. Additionally, the eIDAS server was used to sign the DCCs used in the demonstrator. The purpose of the digital signatures is to prove the authenticity and integrity of the files when the data is needed. The implementation of the service was developed based on examples that are available at Digital Signature Services (DSS) GitHub repository that is based on the development carried out in eSignature initiative that aims to accelerate the use of legally valid electronic signatures in member states of the European Single market [11]. For XML files the eIDAS compliant signature format is XML advanced electronic signatures (XAdES) [12]. A detailed description and examples of an XAdES signature can be found on the website of the World Wide Web Consortium (W3C) [13].

DOI 10.5281/zenodo.5522855

Basic information of digital signatures can be found in the SmartCom deliverables D4 [14] and D6 [15].

3.5 Database API

A database API is used for storing the measurement files that have been signed. Before the data are stored to a structured query language (SQL) database, the measurement file is sent to IOTA where information of the measurement file is stored into a distributed ledger. When the IOTA transaction is completed, a transaction hash that is a digital fingerprint of all the transaction fields is obtained and can be stored in the database. More information about DLTs and IOTA can be found respectively from [16] and [17]. The IOTA implementation counters the following attacks:

- Modifying a record: a modified record will not be found in the valid IOTA transactions when reading from the database
- Adding or removing a record: the number of records in IOTA and database do not match
- Replacing an existing record with a new one: the new record will not be found in valid IOTA transactions.
- Adding an IOTA transaction with the correct IOTA transaction tag but a nonsensical message, i.e., the decrypted message of the added transaction does not follow the defined format, so the transaction is not used for confirming measurements in the database

Once the IOTA transaction has been completed, the database API stores the data into an SQL database. The stored data consists of the following parameters:

- Crane name/identifier

DOI 10.5281/zenodo.5522855

- Gross weight of the container
- Tare weight of the container
- Crane trolley position
- Crane hoist position
- Crane bridge position
- Time stamp
- Container identification
- Fingerprint of the measurement XML file
- Fingerprint of the IOTA transaction
- Cryptographic identifier of the DCC of the measurement instrument used.

In this case the gross weight of the container is the most important one of the measurement results so the DCC of the crane's load sensor is used. The definition of the used cryptographic identifier format is presented in Annex 5.2.

DOI 10.5281/zenodo.5522855

4 User interface

The UI of the demonstrator was developed using React Native that is an open-source framework that allows developers to React and JavaScript for developing UI software for several operating systems [18]. The UI consists of two main views:

- View for the crane operator for creating measurements
- View for users who want to inspect the information of specific containers

In addition to the main functions of the views, the users can also inspect which instruments have been used to execute the measurements.

4.1 Operator view

The operator view displays the data that is fetched from the crane's OPC UA interface. To create a measurement the user must type the ID of the container that is being handled into the text field. After that the user can start the measurement file creation process. Figure 2 shows the operator view of the UI where the steps of the measurement file creation process are also presented.

DOI 10.5281/zenodo.5522855

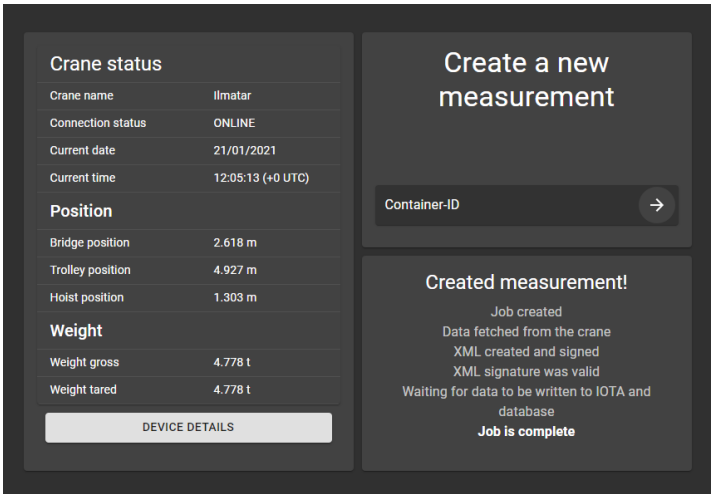

Figure 2. Operator view of the demonstrator UI.

4.2 Container search view

In the container search view the user can search for measurement of a specific container. The measurement events matching the given container ID are listed in the UI based on their time stamps. The search view contains a validations tool that verifies that the weight results of the containers are consistent, the signatures are valid, and the IOTA transaction hash of the measurement event matches the IOTA ledger.

DOI 10.5281/zenodo.5522855

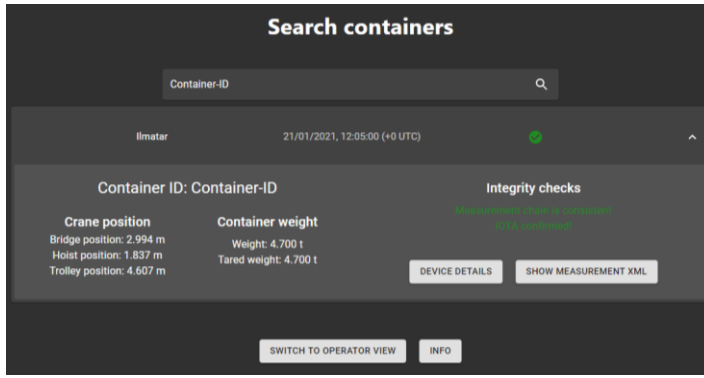

Figure 3. UI showing a valid measurement event.

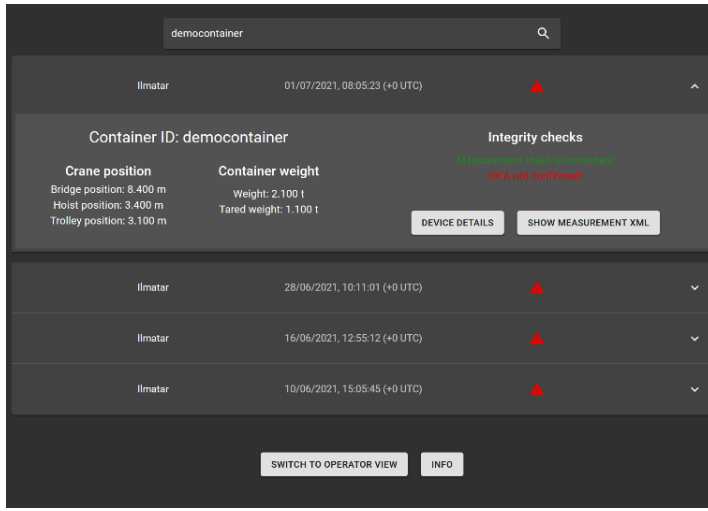

Figure 4. UI showing a measurement event with invalid IOTA validation result.

DOI 10.5281/zenodo.5522855

4.3 Measurement devices

In the measurement devices view the user can inspect the information of individual measurement instruments in the crane system that were used to obtain the measurement results. The information includes the manufacturer and model information obtained from the DCC. Additionally, a cryptographic identifier can be included as well as a validity period or issuing date of the DCC if these are defined in the DCC file.

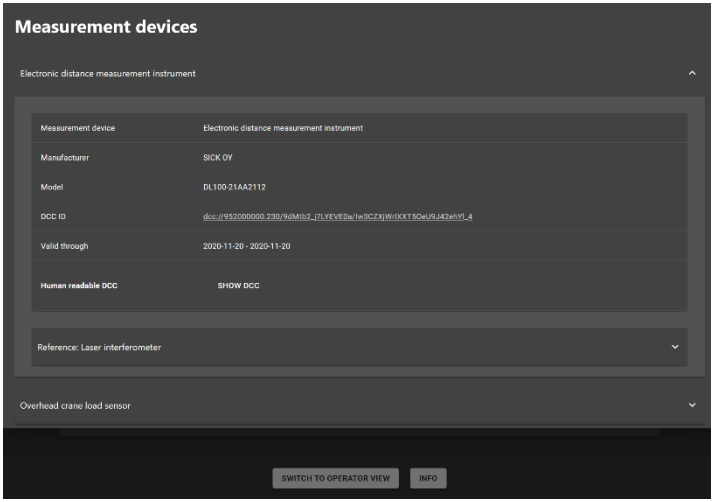

Figure 5. Visualisation of the measurement instrument information.

Clicking the “show DCC” text allows the user to view the DCC of that instrument in a human readable format. The DCC viewer was implemented using JavaScript and extensible stylesheet language transformation (XSLT).

Figure 6, Figure *7* and Figure *8* show how the different parts of the DCC, i.e., the administrative data, measurement

DOI 10.5281/zenodo.5522855

results and signature of the DCC are respectively visualised in the UI. Some of the DCC's content are not fully shown in the viewer when it is first opened but they can be opened to show the content in more detail.

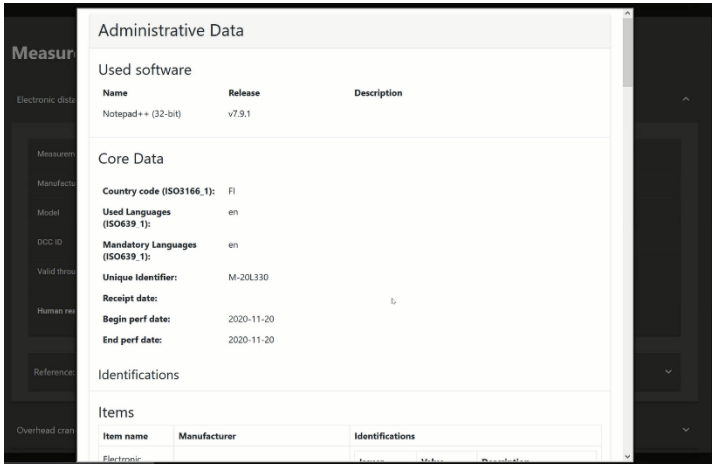

Figure 6. Visualisation of the administrative data of the DCC.

The measurement results are shown in a graph where the horizontal axis represents the individual measurement points in the order they are listed in the DCC, and the vertical axis represents the numerical values of the measurement results. The units of the values are included in the labels of the graph. In addition to the graph the data are also shown in a table.

DOI 10.5281/zenodo.5522855

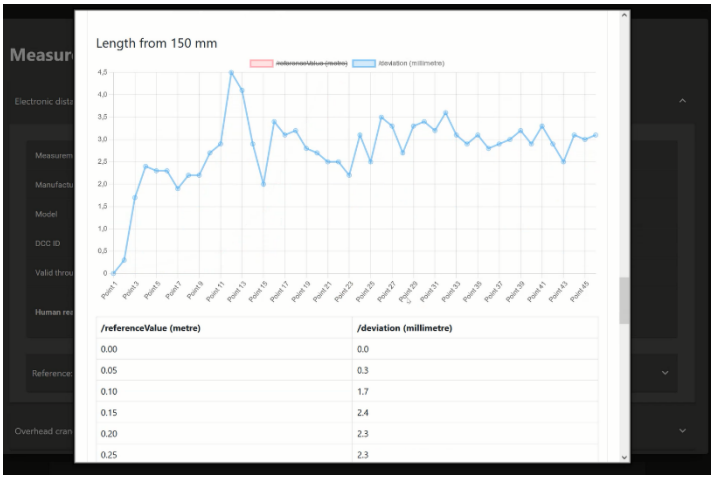

Figure 7. Visualisation of the measurement data of the DCC.

The visualisation of the signature includes a validation feature for validating that the DCC is a real document issued by a trustworthy organisation. Below the validation button there is table where information about signature is shown including the issuer of the key certificate of the key used for signing the DCC, the owner of that specific key, which kind of a key is in question and the fingerprint of the signature. For a demonstration, also example DCCs of the references used in the calibration have been uploaded have been included to the system allowing the visualisation of a part of the calibration chain.

DOI 10.5281/zenodo.5522855

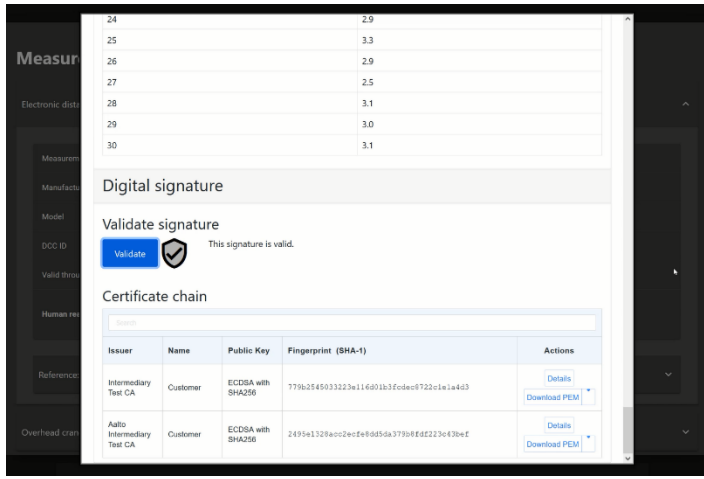

Figure 8. DCC signature validation.

DOI 10.5281/zenodo.5522855

5 Summary

In this report we presented the concept and implementation of a system for secure exchange of mass and position data of containers in harbours as a part of a logistics chain. The approach for ensuring security and trustworthiness of the data included two aspects. Firstly, the correct representation and traceability of the measurement data, and the reliability of the devices used to collect the data were ensured with the use of the D-SI and DCCs for presenting the measurement and calibration data. Secondly, the integrity and authenticity of the data were secured using digital signatures while the database was protected against adding, removing, or replacing data files afterwards using IOTA. By combining these methods and technologies the users can rely on the validity of the data even if the parties exchanging the data do not have an existing mutual trust relationship.

DOI 10.5281/zenodo.5522855

6 References

[1]“Publishable Summary for 17IND02 SmartCom Communication and validation of smart data in IoT-networks.” https://www.ptb.de/empir2018/fileadmin/documents/empir/SmartCom/documents_for_download/SmartCom_17IND02_PublishableSummary.pdf (accessed Aug. 13, 2020).

[2]T. Wiedenhöfer, D. Hutzschenreuter, I. Smith, and C. Brown, “Document describing a universal and flexible structure for digital calibration certificates (DCC),” Nov. 2019, doi: 10.5281/zenodo.3696567.

[3]S. Hackel, F. Härtig, J. Hornig, and T. Wiedenhöfer, “The Digital Calibration Certificate,” *PTB-Mitteilungen*, vol. 127, no. 4, pp. 75–81, 2017, doi: 10.7795/310.20170403.

[4]D. Hutzschenreuter *et al.*, “SmartCom Digital System of Units (D-SI) Guide for the use of the metadata-format used in metrology for the easy-to-use, safe, harmonised and unambiguous digital transfer of metrological data,” Nov. 2019, doi: 10.5281/zenodo.3522631.

[5]“International Maritime Oraganization, Verification of the gross mass of a packed container.” http://www.imo.org/en/OurWork/Safety/Cargoes/Containers/Pages/Verification-of-the-gross-mass.aspx (accessed Mar. 05, 2020).

[6]“Industrial Internet Campus | Aalto University.” https://www.aalto.fi/en/aiic (accessed Sep. 27, 2021).

[7]J. Autiosalo, “Platform for industrial internet and digital twin focused education, research, and innovation: Ilmatar the overhead crane,” in *2018 IEEE*

4th World Forum on Internet of Things (WF-IoT), Feb. 2018, pp. 241–244. doi: 10.1109/WF-IoT.2018.8355217.

[8]"Aalto Smartcom / Ceracrane," *GitLab*. https://gitlab.com/aalto-smartcom/ceracrane (accessed Jan. 06, 2021).

[9]I. Smith, Y. Lou, L. Heindorf, B. Müller, D. Hutzschenreuter, and S. Schönhals, "Good practice guides SmartCom validation," Jul. 2020, doi: 10.5281/zenodo.3816696.

[10] *Regulation (EU) No 910/2014 of the European Parliament and of the Council of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC*, vol. 257. 2014. Accessed: Jan. 22, 2021. [Online]. Available: http://data.europa.eu/eli/reg/2014/910/oj/eng

[11] *Demonstrations for DSS : Digital Signature Service*. AaltoSmartCom, 2020. Accessed: Sep. 21, 2021. [Online]. Available: https://github.com/AaltoSmartCom/dss-demonstrations

[12] ETSI, "XML Advanced Electronic Signatures (XAdES)," Jan. 06, 2009. https://www.etsi.org/deliver/etsi_ts/101900_101999/101903/01.04.01_60/ts_101903v010401p.pdf (accessed Dec. 06, 2020).

[13] "XML Advanced Electronic Signatures (XAdES)." https://www.w3.org/TR/XAdES/ (accessed Dec. 06, 2020).

[14] P. Nikander *et al.*, "Document specifying rules for the secure use of DCC covering legal aspects of

metrology.” Feb. 12, 2020. doi: 10.5281/zenodo.3664211.

[15] T. Mustapää *et al.*, “Guideline describing the concept of UniTerm and how to establish secure communication interfaces in legal metrology,” Jul. 2021, doi: 10.5281/zenodo.5121620.

[16] P. Nikander, J. Autiosalo, and S. Paavolainen, “Interledger for the Industrial Internet of Things,” in *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, Jul. 2019, vol. 1, pp. 908–915. doi: 10.1109/INDIN41052.2019.8972167.

[17] “IOTA Documentation.” https://docs.iota.org/ (accessed Sep. 27, 2021).

[18] “Introduction · React Native.” https://reactnative.dev/docs/getting-started (accessed Sep. 27, 2021).

[19] “Global Document Type Identifier (GDTI) | GS1.” https://www.gs1.org/standards/id-keys/gdti (accessed Sep. 27, 2021).

[20] “DOI® Handbook.” Accessed: Sep. 27, 2021. [Online]. Available: https://www.doi.org/hb.html

[21] “Uniform Resource Identifier (URI): Generic Syntax” https://datatracker.ietf.org/doc/html/rfc3986 (accessed Sep. 27, 2021).

7 Annex

7.1 Example of the used measurement file XML format

```
<?xml version="1.0" ?>
<ev:measurementEvent
xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
xsi:schemaLocation=http://NamespaceTest.com/measuremen
tEvent
xmlns:si=https://ptb.de/si
xmlns:ev=http://NamespaceTest.com/measurementEvent
xmlns:m=http://NamespaceTest.com/measurementMetadata
>
  <ev:metadata>
     <m:craneName>Crane 1</m:craneName>
     <m:dateTime>2021-06-30T09:30:10Z</m:dateTime>
     <m:containerId>34ecd299-7f76-4972-bd90-
     9e0615fe5e4e</m:containerId>
  </ev:metadata>
  <ev:measurement>
     <si:real>
        <si:label>load_tared</si:label>
        <si:value>122</si:value>
        <si:unit>\kilogram</si:unit>
        <si:expandedUnc>
           <si:uncertainty>0.50</si:uncertainty>
           <si:coverageFactor>2</si:coverageFactor>
           <si:coverageProbability>0.95>
           </si:coverageProbability>
           <si:distribution>normal</si:distribution>
        </si:expandedUnc>
     </si:real>
     <si:real>
        <si:label>load_gross</si:label>
        <si:value>123</si:value>
```

DOI 10.5281/zenodo.5522855

```
      <si:unit>\kilogram</si:unit>
      <si:expandedUnc>
         <si:uncertainty>0.50</si:uncertainty>
         <si:coverageFactor>2</si:coverageFactor>
         <si:coverageProbability>0.95
         </si:coverageProbability>
         <si:distribution>normal</si:distribution>
      </si:expandedUnc>
   </si:real>
   <si:real>
      <si:label>hoist_position</si:label>
      <si:value>122</si:value>
      <si:unit>\metre</si:unit>
      <si:expandedUnc>
         <si:uncertainty>0.50</si:uncertainty>
         <si:coverageFactor>2</si:coverageFactor>
         <si:coverageProbability>0.95
         </si:coverageProbability>
         <si:distribution>normal</si:distribution>
      </si:expandedUnc>
   </si:real>
   <si:real>
      <si:label>trolley_position</si:label>
      <si:value>122</si:value>
      <si:unit>\metre</si:unit>
      <si:expandedUnc>
         <si:uncertainty>0.50</si:uncertainty>
         <si:coverageFactor>2</si:coverageFactor>
         <si:coverageProbability>0.95
         </si:coverageProbability>
         <si:distribution>normal</si:distribution>
      </si:expandedUnc>
   </si:real>
   <si:real>
      <si:label>bridge_position</si:label>
      <si:value>122</si:value>
      <si:unit>\metre</si:unit>
```

DOI 10.5281/zenodo.5522855

```
         <si:expandedUnc>
            <si:uncertainty>0.50</si:uncertainty>
            <si:coverageFactor>2 </si:coverageFactor>
            <si:coverageProbability>0.95
            </si:coverageProbability>
            <si:distribution>normal</si:distribution>
         </si:expandedUnc>
      </si:real>
   </ev:measurement>
</ev:measurementEvent>
```

DOI 10.5281/zenodo.5522855

7.2 Definition of the cryptographic identifier format

The identifier format used for the DCCs is based on the Global Document Type Identifier (GDTI) [19], Digital Object Identifier (DOI) [20] and Uniform Resource Identifier (URI) standards [21]. The format is described in detail in Figure 9.

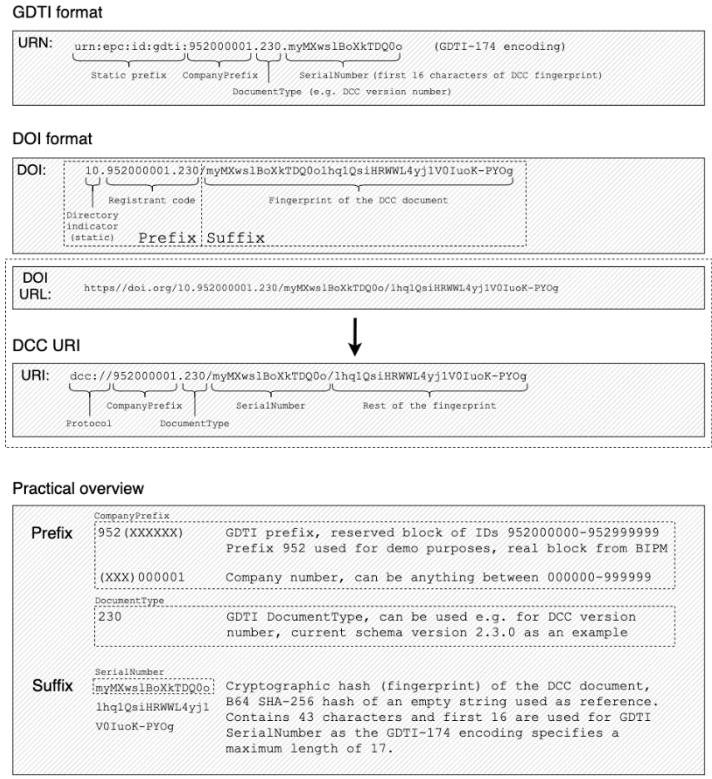

Figure 9. Definition of the cryptographic identifier format.

DOI 10.5281/zenodo.5522855

The content presented was developed within the framework of the EU-funded project SmartCom *"Communication and validation of smart data in IoT-networks" with the* support of international partners from science and industry.

https://www.ptb.de/empir2018/smartcom
(retrieved February 2020)

The EMPIR initiative is co-funded by the European Union's Horizon 2020 research and innovation programme and the EMPIR Participating States

DOI 10.5281/zenodo.5522855