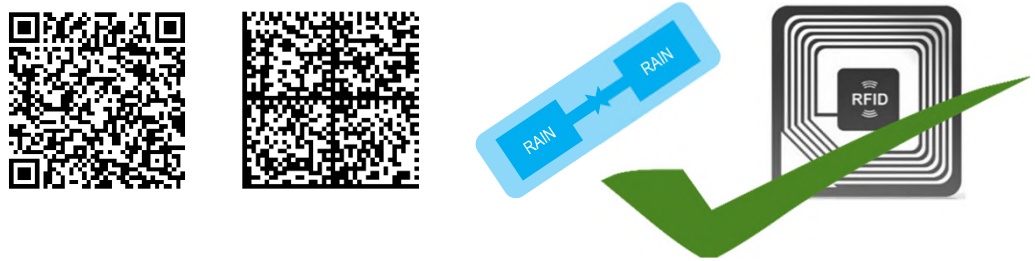
# An introduction to the ISO/IEC 20248 Data Structure (DigSig) using RAIN EVI

RAIN Seattle - July 2017

Bertus Pretorius – apretorius@licensys.com, b.Pretorius@toennjes.com

# What is ISO/IEC 20248?



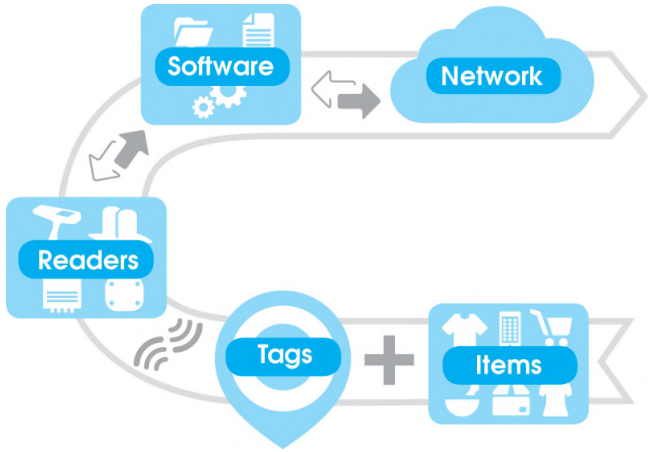

ISO/IEC 20248 specifies a method to specify verifiable data structures.

It provides verifiable unique numbers with attributes and context in all languages.
It uses standard X.509 PKI digital signatures (in other words “server certificates”).
The Certificate specifies the data owner, data structure and the data description.
It also contains all static data.
The data carrier only contains the variable data and the signature called a DigSig.

It provides full interoperability between RAIN, NFC and barcodes.
It is fully interoperable with the EPC Tag Data Standard.

20248 enables automated identification in IoT in the same way HTML enabled the WWW.

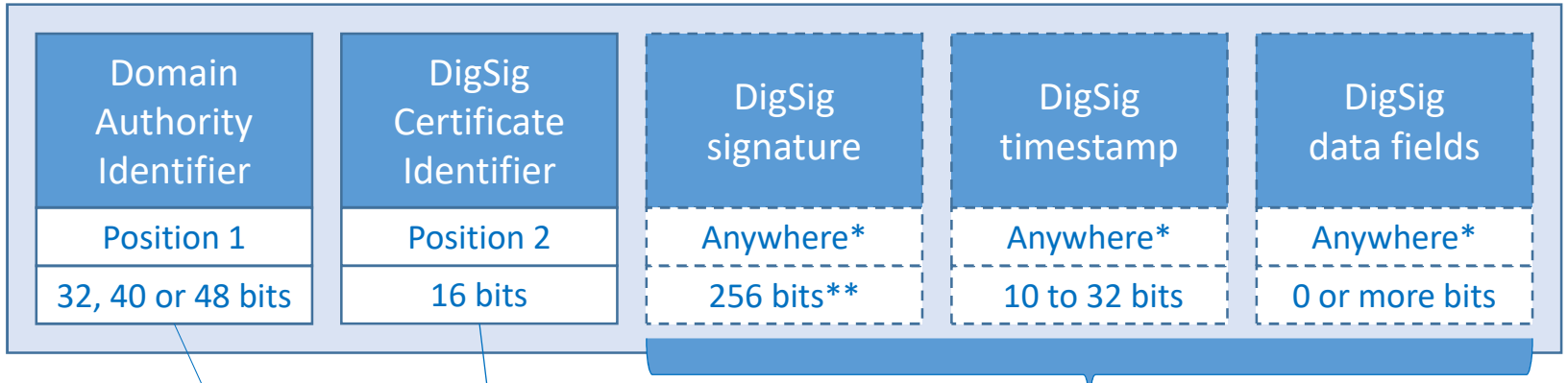
# 20248 is the “HTML” of automated identification




| | | |
|---|---|---|
| **Data owner** | Web server owner | Domain Authority, e.g. the transport department |
| **Data definition** | HTML framework | PKI Certificate with the DigSig Data Description |
| **Data carrier** | HTTP, HTTPS | RAIN, RAIN-Crypto, NFC and barcodes |
| **Data** | Secure HTML | DigSig (Certificate reference, data elements & signature) |
| **Display module** | Web Browser | 20248 enabled reader |
| **Data input/output** | JSON/XML | JSON |

*Images from www.ipzonecomputer.com and RAINrfid.org*

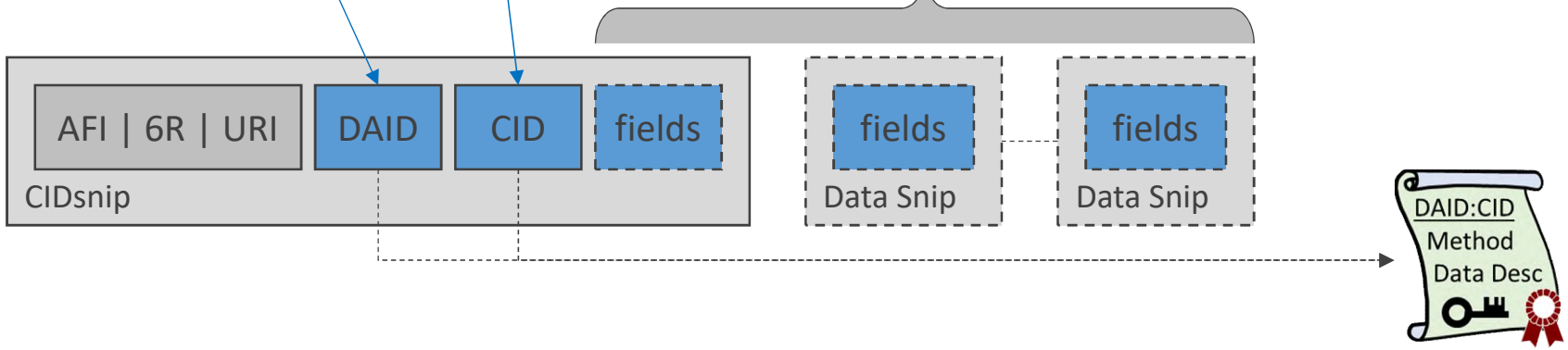
# Anatomy of an encoded DigSig




A DigSig is stored as one or more Snips, each Snip maps to a part of an AIDC data carrier memory page.
A Snip is a continues stream of bits.
The first Snip to be read is called the CIDSnip.

- It shall start with the Domain Authority Identifier (DAID) and the DigSig Certificate Identifier (CID).
- It typically contains data fields.

Data Snips only contains data fields.




The DigSig data construct

Continues part of an AIDC data carrier memory page.

**Data owner specified**
The dotted outline indicates the element is data-owner specified and may be omitted.

***Anywhere**
Anywhere from field position 3. The {DAID || CID} references the DigSig certificate which describes the data structure; field position, type, storage device, encoding and multi-language description.

****Signature bits**
The signature size is dependent on the security use case. It can be:

- zero (no verification)
- 16-32 bits (CRC level verification)
- 80-255 bits: (weak signature)
- 256 bits (current recommendation)
- > 256 bits (very strong)

# Tönnjes (and LicenSys)

- The Tönnjes group provides vehicle licence plates in 132 countries.
- LicenSys, a member of the Tönnjes group, manufactures all the plates for Australia and New Zealand.

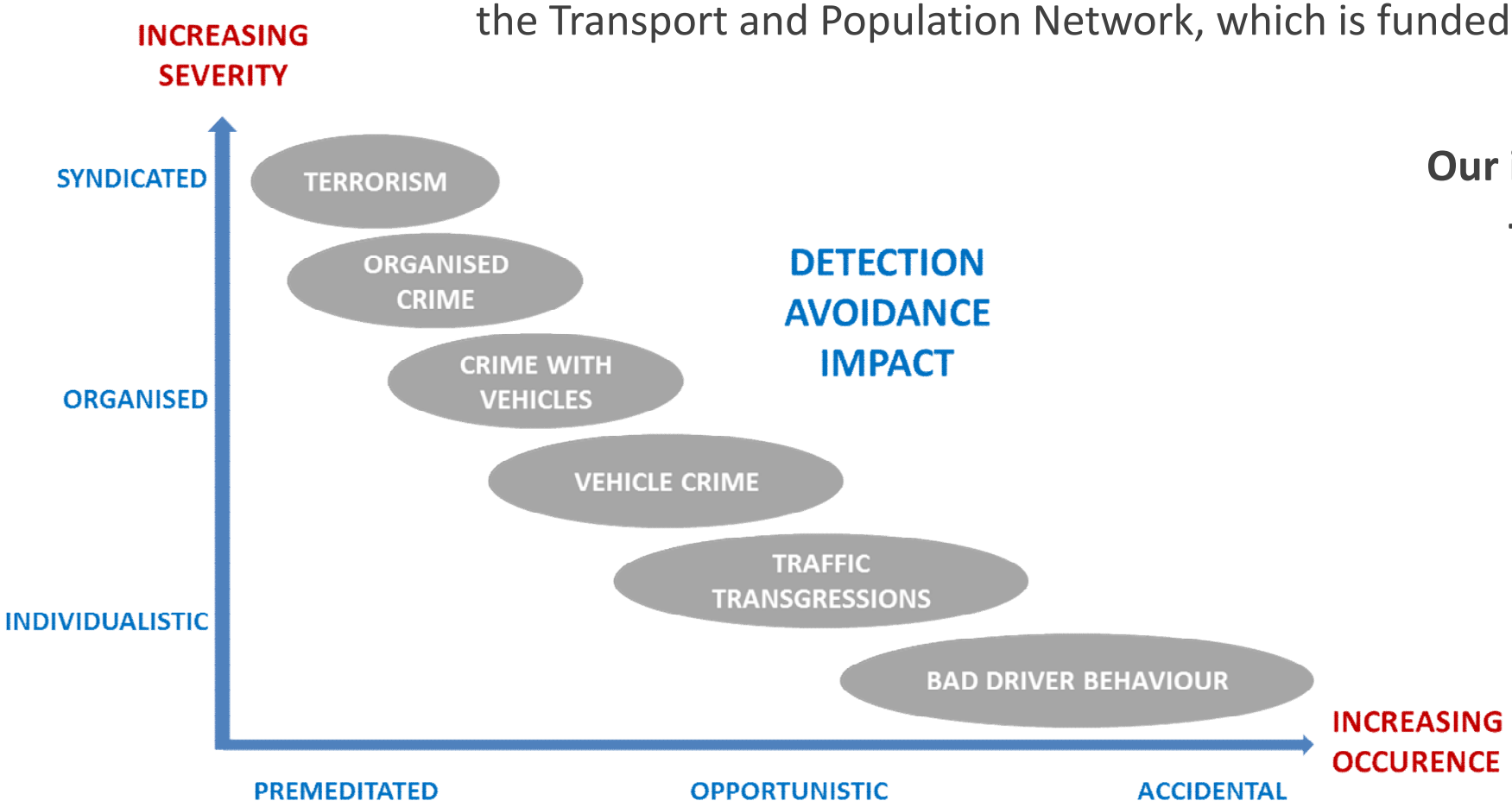
- We have noticed a global and significant discrepancy between plates manufactured, vehicle registrations and reported vehicles on the road.
- Currently no jurisdiction has the ability to verify their vehicle fleet.
- We strongly believe that all vehicles must be identifiable by humans and machines; **the vehicle licence plate is the passport of the vehicle**.

- Imaging can not tell if a plate is genuine.  RAIN rfid can!

# The collaborative fleetVALID research project

www.fleetVALID.com is a University of Queensland Project.

It is funded by the state and federal government. It forms part of a bigger research project, the Transport and Population Network, which is funded by the Treasury and Ports Authority.




**Our interest is the DETECTION AVOIDER.**

**This "1%" is "80%" of our problems.**

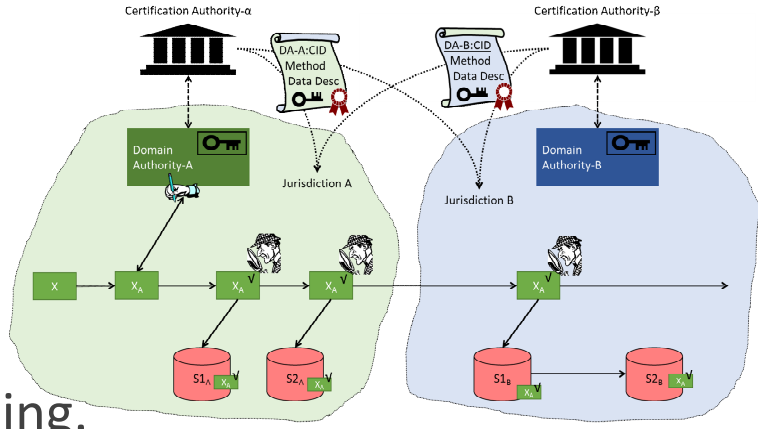
**We aim to detect:**

- Vehicle identity anomalies
- Driver behaviour anomalies
- Vehicle fitness anomalies.

**for immediate intervention.**

**Automated policing**
**Manual intervention**

# DigSigs are a key enabler for positive vehicle identification




- Jurisdictions have difficulty to agree on data structures and encoding.
  - ✓20248 provides interoperability on a sematic level. They only need to agree that a VIN is a vehicle identification number, the licence number appears on the plate, vehicle colour and shape is just that…
  - ✓20248 automates encoding and is therefore transparent to the integrator.
  - ✓20248 provides for backwards compatibility.

- Jurisdictions, for security and cost reasons, do not want to share their data bases.
- Connectivity is expensive and often not available (e.g. just out of town in Australia).
  - ✓20248 provides for off-line verification. For example, a German policeman can verify a Russian vehicle licence without the need to connect to the Russian data base.
    All he needs is the Russian DigSig Certificate, which is on its own, like a server certificate, verifiable and linked to the issuer.

# DigSigs are a key enabler for positive vehicle identification

- Limited time to read the data in free flow applications.
  - ✓ 20248 compacts data optimally.
  - ✓ Static data is stored in the certificates.
  - ✓ 20248 allows the decoding of partially read data.
  - ✓ 20248 provides a method to create independent records on a single tag. The free-flow data is stored in the primary record.

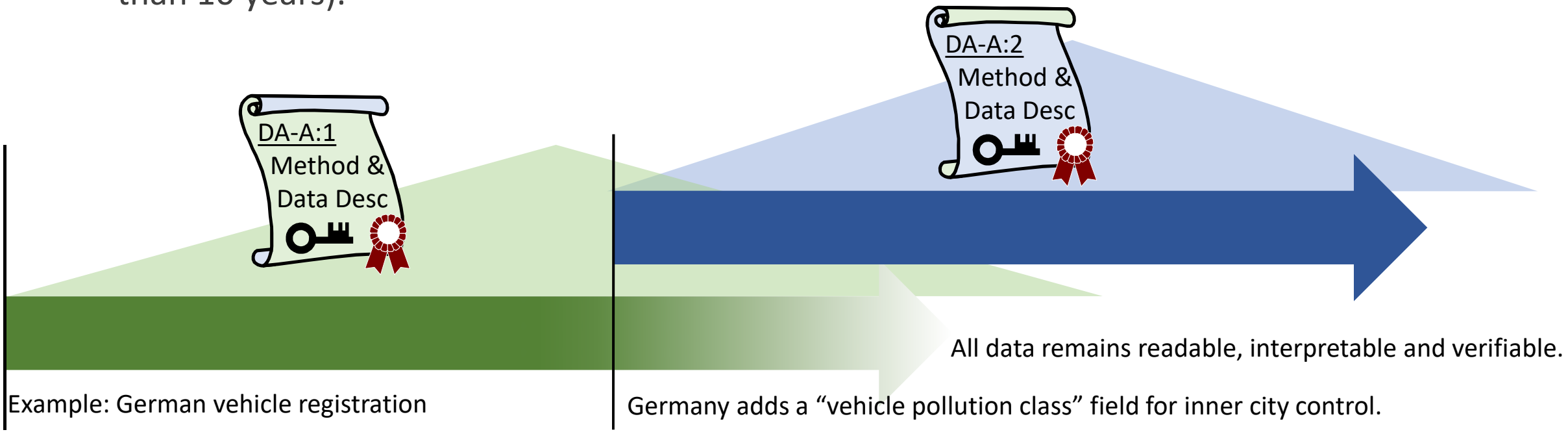
- Some jurisdictions want crypto tags.
  - ✓ 20248 supports key management for crypto tags.

- Some jurisdictions want encrypted data (on and off the tag).
  - ✓ 20248 provides for private containers and the key management for such containers. The closed containers are verifiable using the digital signature.

# DigSigs are a key enabler for positive vehicle identification

- All jurisdictions need revocation
  - ✓20248 provides revocation of individual tags and data structures.

- All jurisdictions need to change vehicle data structures (note average vehicle lifespan is more than 10 years).



# DigSigs are a key enabler for positive vehicle identification

- Some jurisdictions can not afford more than 96 bit tags.
  - ✓ The 96 bit RAIN tag uses a *null encryption* DigSig containing only the plate number. This is used to identify the vehicle in free-flow use-cases.
  - ✓ A barcode on the windscreen label contains a DigSig barcode encoding the registration information, including an expiry date. It also points to the RAIN tag TID creating a method to detect barcode copying. This is verified in road bocks and police stops. Such a barcode can easily be renewed yearly.
  - ✓ In pharma the RAIN tag provides for traceability and the QR for verification using a mobile phone.



Verification success

Certificate information

| | |
|---|---|
| Issued by: | dauri1.20248.info |
| CN: | dauri1.20248.info/cid/424 |
| Expires on: | Tue Jul 10 10:00:00 GMT+10:00 2018 |
| Sig scheme: | ECBNwithSHA256 |

Information

| | |
|---|---|
| Spec version: | ISO/IEC CD2 20248:2016 |
| DAURI: | https://dauri1.20248.info |
| DAID: | QC DGSG |
| CID: | 424 |
| Created at: | 2017-07-10T20:16 |

ABC Pharma

Data fields

Product name
JOLLIPIDEM

Product mark
JPM01

Pack size
10

Medication type
Coated tablet

Demonstration TID - use 4 chars e.g. ABCD
abcd

Demonstration UII/EPC - use 4 chars e.g. ABCD
abcd

# DigSigs are a key enabler for positive vehicle identification

- Some jurisdictions want to retro-fit a DigSig RAIN tag on vehicles and containers.
- Industry want to use the official vehicle identification DigSigs in their services.
  - ✓ 20248 supports the referencing of other standard data carriers.
  - ✓ 20248 supports the referencing of 20248 data structures in other carriers. This is useful, for example, cross-border documents which links a vehicle and trailer with a shipping container and RAIN eSeal. This linking DigSig may be stored in a QR code. There must be at least one TID verifiable RAIN tag in the group to detect copying.


# RAIN ISO examples: 96 bit UII + TID + User Mem

*In this case the unique portion of the TID is used as the unique number. On-chip crypto graphic protection of the TID can be used to guarantee its uniqueness. Note the DigSig Certificate support key management for on-chip crypto.*

| PC Bits (MB01) | | | | | UII (MB01) | | | TID (MB10) |
|---|---|---|---|---|---|---|---|---|
| UII Len | User Mem | XI | Toggle EPC\|ISO | AFI | DAID | CID | [Company assigned fields] | |
| 00110 | 0 | 0 | 1 | 0x92 | 32, 40 or 48 bits | 16 bits | 64, 56 or 48 bits | 32 to 96 bits |

*The above case provides data owner and data structure verification.*

*The below case also provides for data verification.*

| PC Bits (MB01) | | | | | UII (MB01) | | | TID (MB10) | User Mem (MB11) |
|---|---|---|---|---|---|---|---|---|---|
| UII Len | User Mem | XI | Toggle EPC\|ISO | AFI | DAID | CID | [Company assigned fields] | | signature, timestamp [Company assigned fields] |
| 00110 | 1 | 0 | 1 | 0x92 | 32, 40 or 48 bits | 16 bits | 64, 56 or 48 bits | 32 to 96 bits | ≥ 256 bits |

*Note, these methods allow for decoding of the data of partial reads in difficult read scenarios since all fields are described by the DigSig Certificate.*

# RAIN EPC examples: 96 bit + TID + User Mem

*By specification, an EPC tag may only store a DigSig in User memory. This is indicate with the User memory bit and DSFID set to 17.*

| PC Bits (MB01) | | | | | EPC (MB01) | TID (MB10) | User Mem (MB11) | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| UII Len | User Mem | XI | Toggle EPC\|ISO | RFU | GS1 EPC | | DSFID | DAID | CID | signature, timestamp [Company assigned fields] |
| 00110 | 1 | 0 | 0 | 0xXX | *96 bits* | *96 bits* | 0x11 | *32, 40 or 48 bits* | *16 bits* | *≥ 0 bits* |

*The above case uses null-encryption. It provides data owner and data structure verification as the DigSig Certificate is verified.*

*The below case also provides for data verification, which includes the EPC and TID, therefor linking additional fields to the EPC and/or TID.*

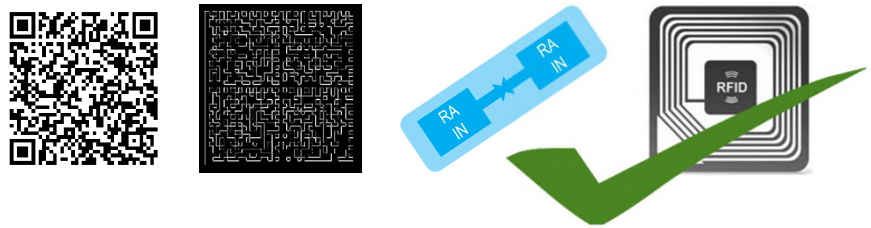
| PC Bits (MB01) | | | | | EPC (MB01) | TID (MB10) | User Mem (MB11) | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| UII Len | User Mem | XI | Toggle EPC\|ISO | RFU | GS1 EPC | | DSFID | DAID | CID | signature, timestamp [Company assigned fields] |
| 00110 | 1 | 0 | 0 | 0xXX | *96 bits* | *96 bits* | 0x11 | *32, 40 or 48 bits* | *16 bits* | *≥ 256 bits* |

# That’s all.  Questions?

Technical bonus slides follow.  Please see the published presentation.

# What is an ISO/IEC 20248 data structure (DigSig)?

- A DigSig is a provable data structure that can be stored on an AIDC data carrier/device.
  - *It is very useful between IoT devices, especially in representing a Thing, and in the creation of Chains of Accountability.*
- A DigSig is inherently (and required to be) unique.
- A DigSig can be revoked.
- A DigSig is issued by the Domain Authority when the Domain Authority digitally signs the DigSigs data.
- The DigSig Data Description (DDD) is defined by a Tag/Data Owner (the Domain Authority). The DigSig Data Description is published in an X.509 Digital Certificate (DigSig Certificate) by the Domain Authority.
  - *The DigSig Data Description and its owner of a DDD is therefore verifiable.*
  - *Many DigSig data descriptions are active at the same time.*
- Anybody with the DigSig Certificate can decode and verify a DigSig.  Partially read DigSigs are decodable.
- A DigSig can be stored in any combination of AIDC devices.  The DigSig AIDC encoding is standardised:
  - RFID: Application Family Identifier ➔ 0x92
  - Barcodes: ASC MH 10.8 Data Identifier ➔ 6R
  - URI: all data carriers which supports URIs can store a DigSig.



# Key aspects

**ISO/IEC 20248**:

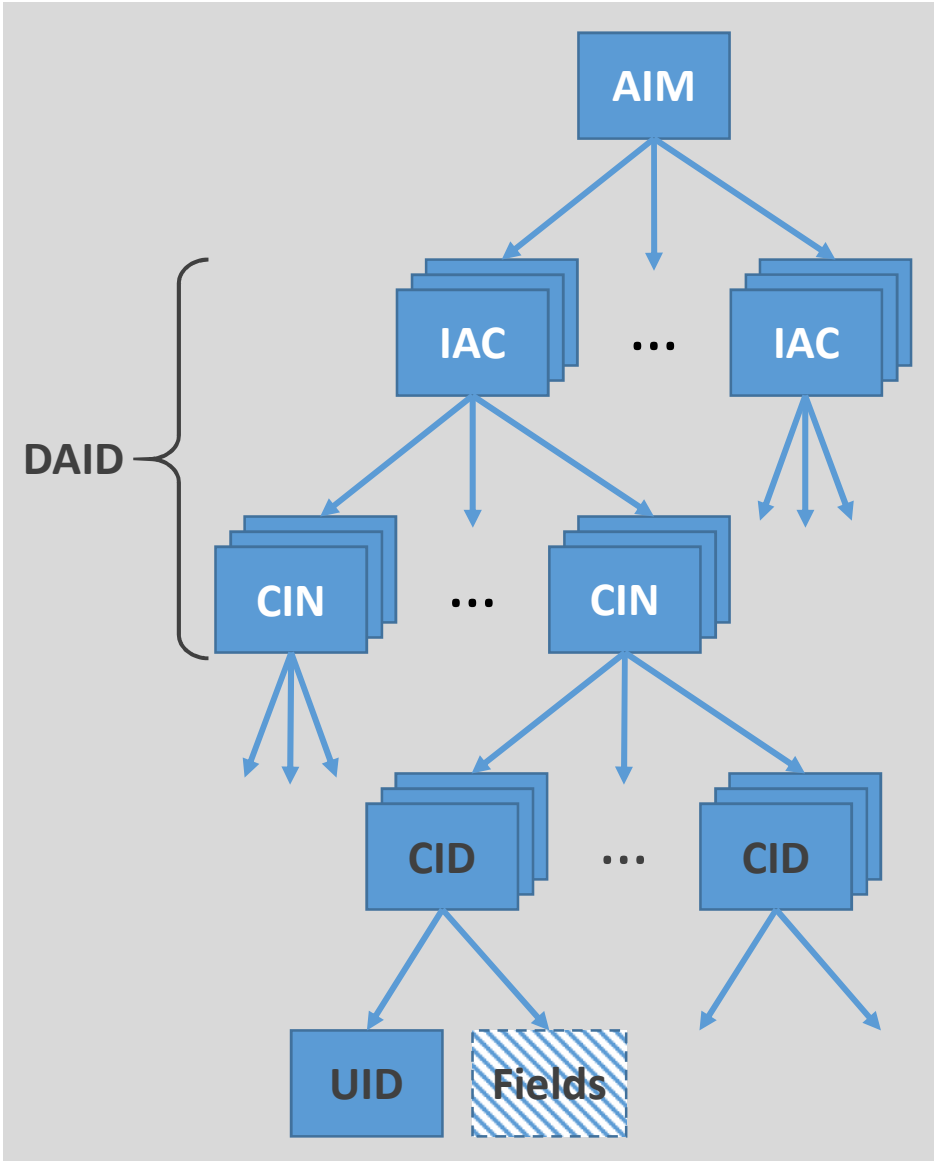
- allows for verification of the data owner, the data structure and the data at the time of reading the tag and after the read event.  The latter supports IoT Edge and Fog schemas.
- supports partial reads, which means that the data can still be decoded even if the full data set is not read.
- supports tags with limited memory (e.g., a 96-bit RAIN tag) by excluding the signature.
- supports revocation.
- supports key management for crypto tags.
- supports Thing attribute fields as defined by the data owner.
- allows for seamless maintenance/change of data structures.

4 interoperable methods for unique numbering is proposed, of which two can be standardised in a manner NOT to require the ISO/IEC 20248 DigSig Certificate to decode.

# ISO/IEC 15459 Company Identification

1. ISO/IEC 15459 is currently in wide use with barcodes.
2. AIMglobal is the ISO/IEC 15459 Registration Authority who assigns IACs.
3. IAC: Issuing Agency Code; Issuing Agencies assign Company Identification Numbers (CINs).
   Current IACs are 0 to 9, A to J, LA to UZ, and KAA to ZZZ.
   Example: “QC” is the Euro Data Council IAC.
   Example: “0” to ”9” are assigned to the GS1 Global Office.
4. Company Identifying Number (CIN)
   CIN = Capital alphanumeric; typically 3 to 6 alphabetic characters, or a number typically up to 9 digits.
   Example: “DGSG” is assigned by “QC” to the 20248.org demo system.
5. ISO/IEC 15459-3 Identifier common rules:
   - Make it as short as possible.
   - Use the Invariant Character Set of ISO/IEC 646: ”0” to ”9” and ”A” to ”Z”.

*DAID and CID are DigSig compulsory fields with UID and fields specified by the data owner .*



# DigSig Domain Authority Identifier [DAID]

**DAID = ISO/IEC 15459 {IAC || CIN}**

The Company is the ISO/IEC 20248 Domain Authority (data/tag owner) who:

a. Specifies the unique number/identifier
b. Specifies optional fields
c. Specifies the DigSig Data Structure to be included in the DigSig Certificate
d. Assigns a unique DigSig Certificate Identifier (CID) for the DigSig Data Structure
e. Publishes the DigSig Certificate, which contains the DigSig Data Description
f. Issues a unique number/identifier
g. Signs and issues DigSigs
h. May revoke DigSigs

**Every <u>DigSig data structure</u> is:**

- Unique,
- Verifiable, and
- Provably linked to the Domain Authority.

**Every <u>DigSig is unique</u> by:**

- Virtue of the digital signature which is inherently (and required to be) unique,
- A company assigned unique number (for example EPCs), or
- The unique portion of an RFID TagID (RAIN tag TID). This unique TagID may be cryptographically provable.

# DAID, CID and UID numbers

- ISO/IEC 15459 makes provision for:
  - 4335 Issuing Agencies.
  - Each Issuing Agency can assign 2,176,782,336 company identification numbers.
  - Each company may have 65,534 live DigSig data structures.
  - Unique live data structures: 618,401,854,388,183,040 with unlimited company assigned unique numbers
- The unique number can be constructed as follows:
  - ➔ DAID || CID || <Company assigned unique number>
  - ➔ DAID || CID || *the unique part of the* TID
  - ➔ DAID || CID || Tag Encrypted TID
  - ➔ DAID || CID || <any of the three UIDs above> || signature

    The DigSig Data Description contained in the DigSig Certificate reference (a verified data structure) by {DAID, CID} informs the reader which method to use.
- The result is a verifiable data owner and data structure, with a verifiable unique number that is specified by the data owner (Domain Authority) according to the use-case and tag-reader capabilities.

# RAIN unique identifier example: 96 bit UII

Only the 96 bit UII is read: the DigSig contains {DAID, CID, <Company assigned unique number [and fields]>}

1. The compulsory fields <signature> and <timestamp> binary format is set to **zero**:
   - The data CAN NOT be verified since the DigSig does not contain a signature.
   - The data structure and data owner IS verifiable since the verifiable DigSig Certificate contains the owner and data structure.
2. When an ISO/IEC 18000-63 tag is read, the following happens:
   a. The AFI indicates that the UII contains a DigSig and the length is 96 bits.
   b. The 96 bit DigSig is read and is passed to the DigSig decoder.
   c. The DigSig decoder decodes the {DAID, CIN} and selects the referenced DigSig Certificate from the OS certificate store.
   d. The DigSig decoder verifies the DigSig Certificate {DAID, CIN, DigSig Data Description}.
   e. The DigSig decoder decodes the <Company assigned unique number> according to the DigSig Certificate and passes the DigSig data back to the application in JSON.

| PC Bits (MB01) | | | | | UII (MB01) | | |
|---|---|---|---|---|---|---|---|
| UII Len | User Mem | XI | Toggle EPC\|ISO | AFI | DAID | CID | Company assigned unique number [and fields] as described by the DigSig Certificate |
| 00110 | 0 | 0 | 1 | 0x92 | 32, 40 or 48 bits | 16 bits | 64, 56 or 48 bits |

*Note: this method works with any length of UII memory.*