

ISO/IEC 20248

An Introduction

Bertus Pretorius – apretorius@licensys.com

Introduction: ISO/IEC 20248

Automatic Identification and Data Capture Techniques – Data Structures – Digital Signature Meta Structure

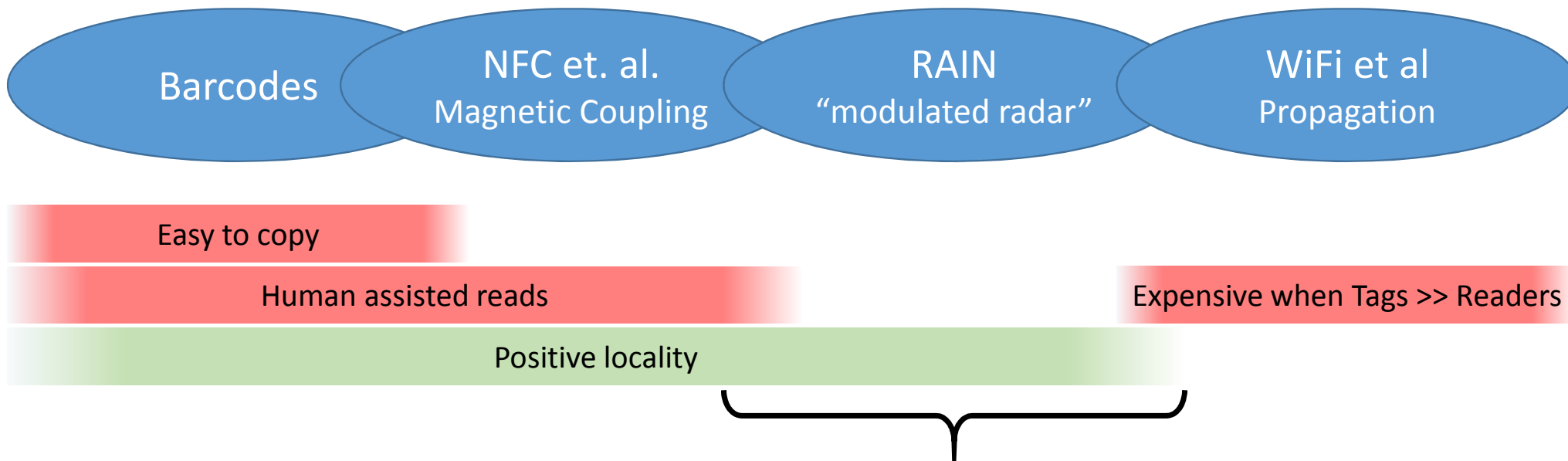
- It is under development by ISO/IEC JTC1 SC31 WG2
It is based on SANS 1368 which is published and in use.
- It specifies a method:
 - whereby the AIDC data is structured and digitally signed, and
 - whereby the read method/process and data interpretation/presentation is defined as a verifiable directive using a X.509 version 3 certificate. It is in fact a X.509 (ISO/IEC 9594-8) application standard.
- The purpose is to:
 - interpret/present and verify data at the read point without the need for an online connection,
 - provide interoperability between carrier media; i.e. QR to RAIN,
 - provide interoperability between data domain authorities; ie. the {Navy, Army...}, DoT{OR, WA...}

The ISO/IEC 20248 data structure is called a "DigSig" which refers to a small, in bit count, digital signature.

The ISO/IEC 20248 also provides an effective and interoperable method to exchange data messages in the Internet of Things [IoT] and machine to machine [M2M] services allowing intelligent agents in such services to authenticate data messages and detect data tampering.

AIDC technologies: where does RAIN fit in?

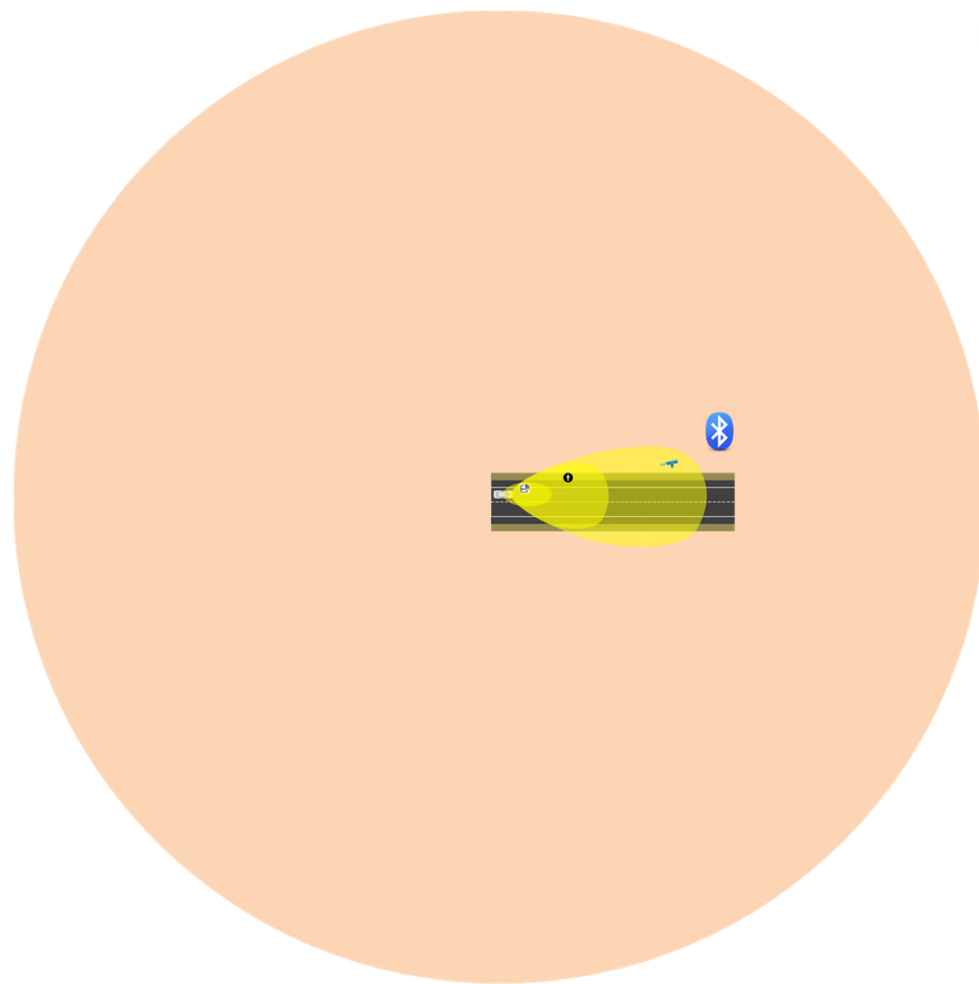
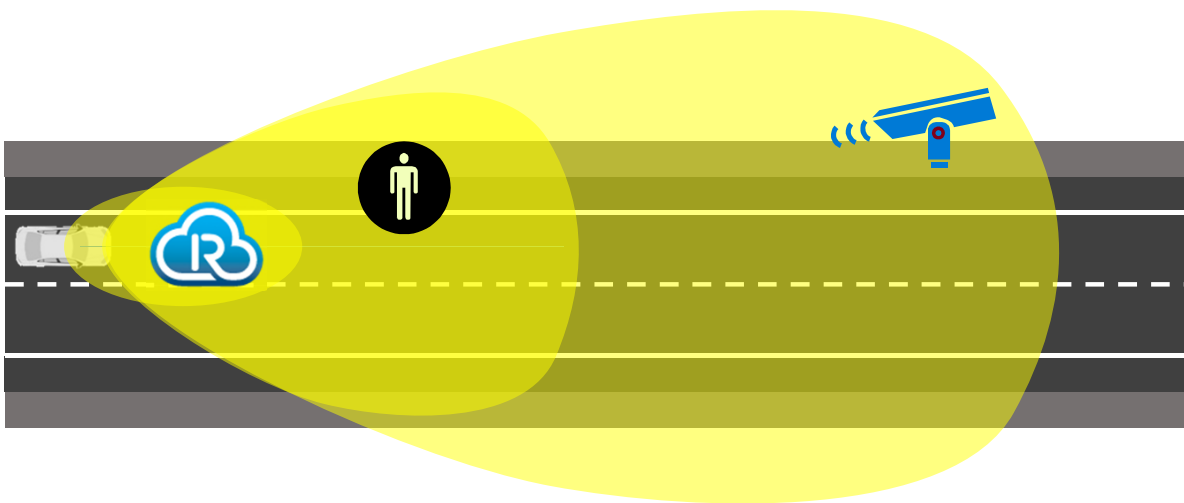
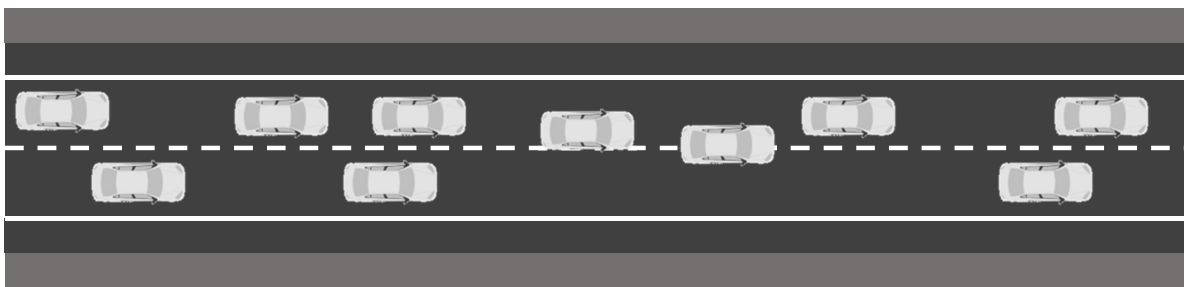
Automated identification data carriers allow information systems to identify and locate real world objects.



RAIN unlocks M2M and IoT; for IoT and M2M to work a machine must be able to **detect, identify and verify**; and then **act**; and when the machine cannot perform its function, call a human to sort out the problem.

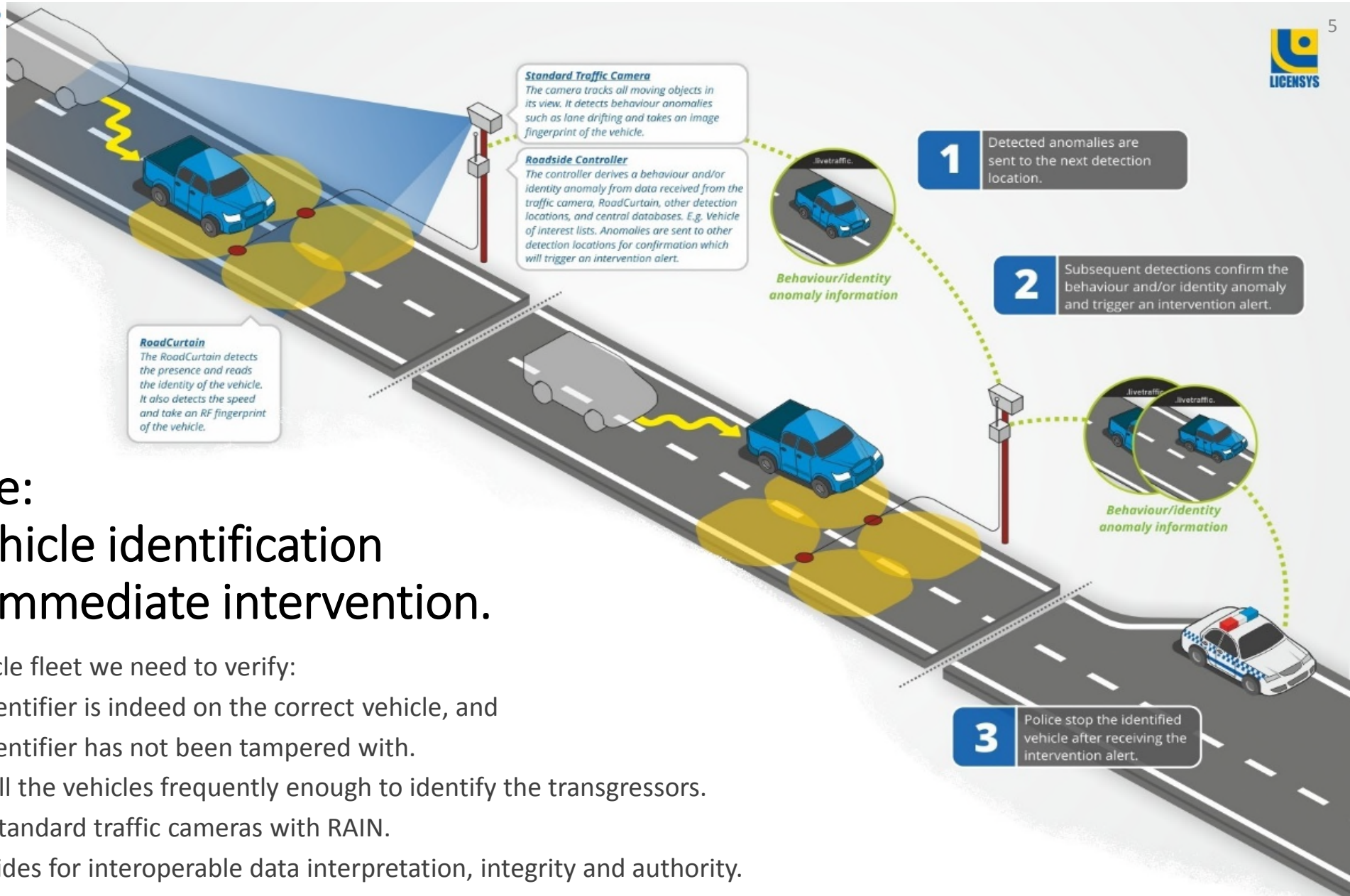
Note: IoT and M2M in the end serve humans.

The read range issue



The read range of RAIN is far enough to lose the human in the read operation, ie. to automate - while maintaining effective locality.

The underlying business case is to know what is missing where.



An example: Positive vehicle identification to enable immediate intervention.

To legalise the vehicle fleet we need to verify:

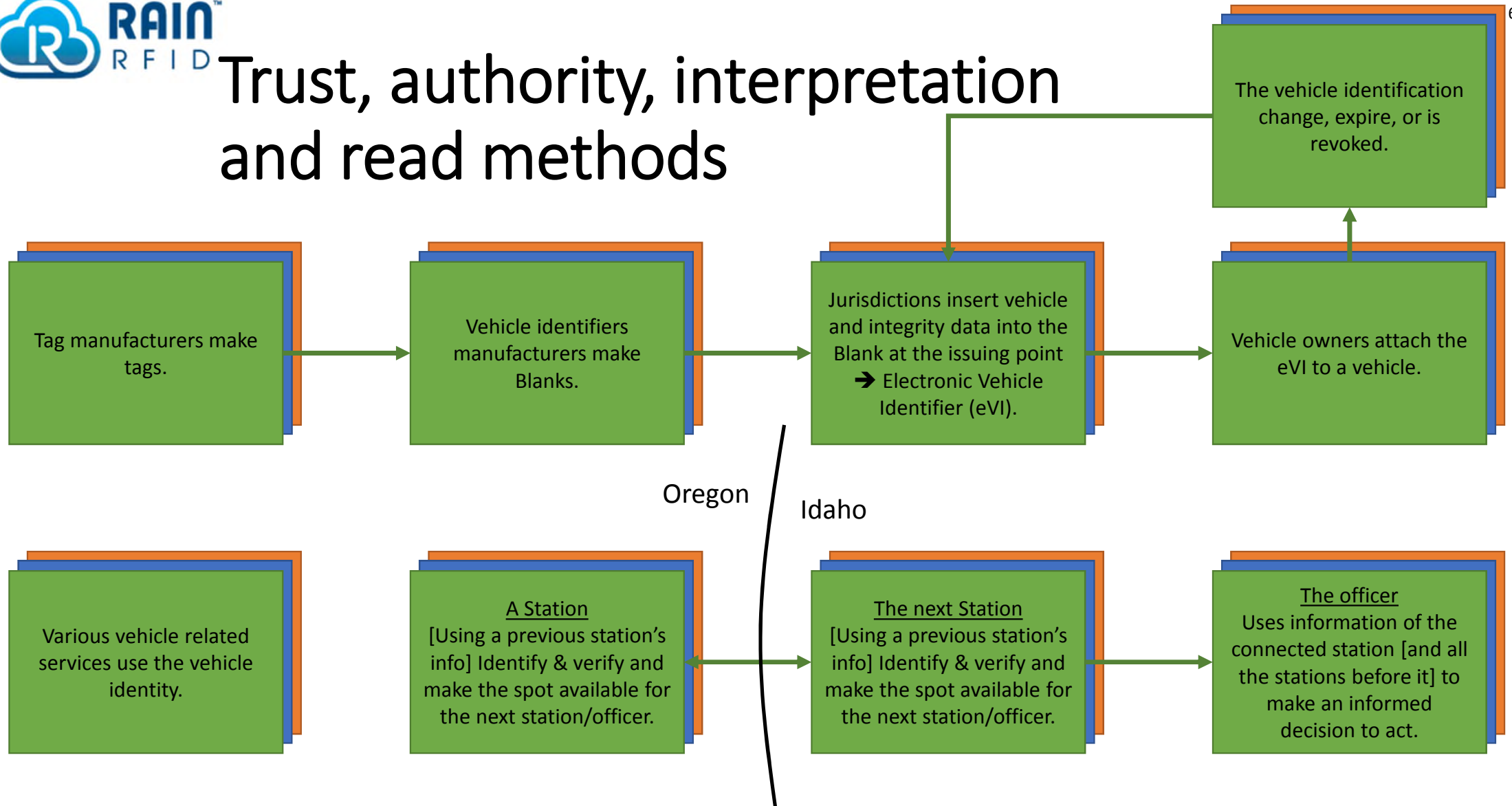
- that the vehicle identifier is indeed on the correct vehicle, and
- that the vehicle identifier has not been tampered with.

We need to verify all the vehicles frequently enough to identify the transgressors.

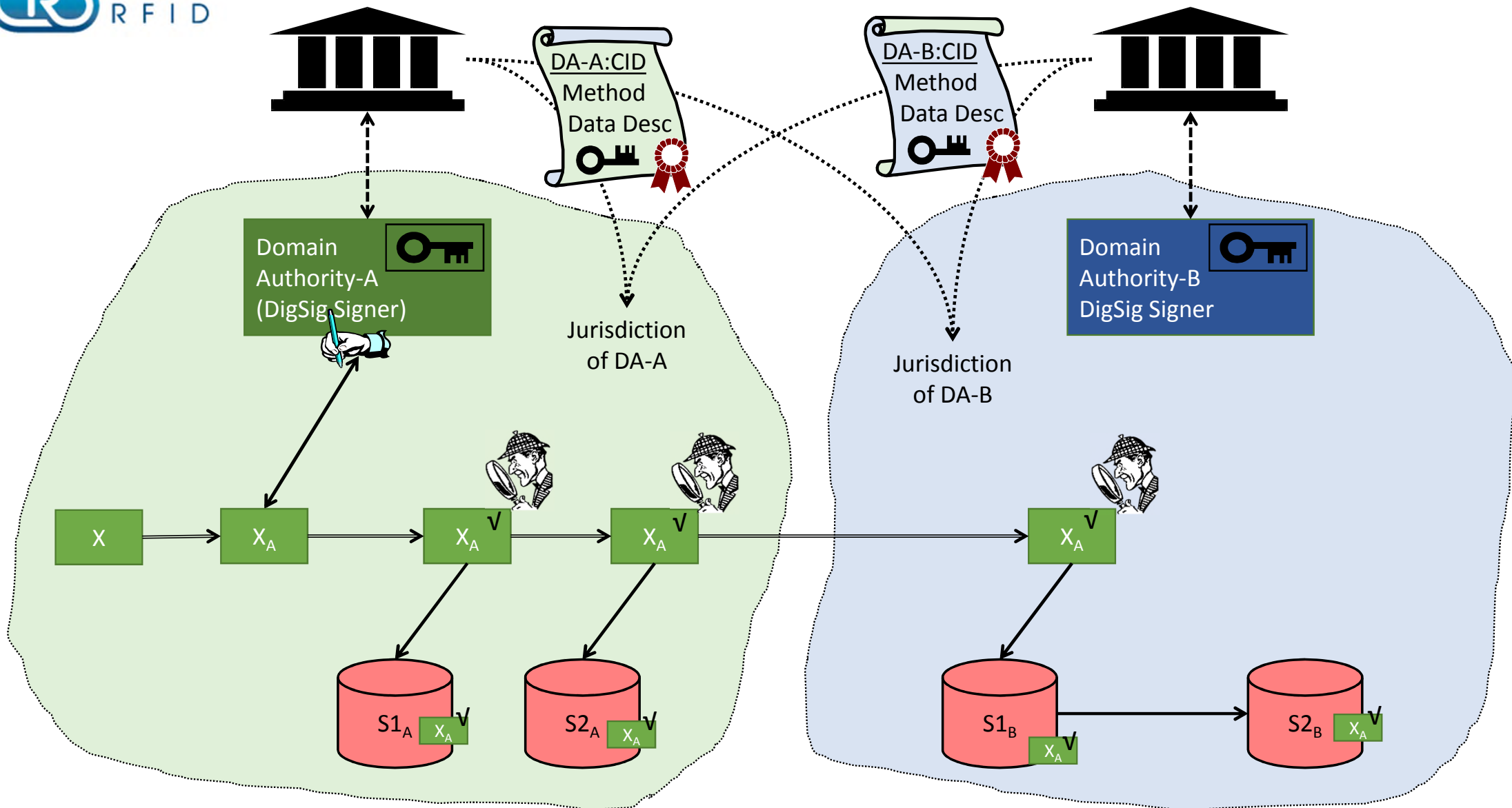
For this we merge standard traffic cameras with RAIN.

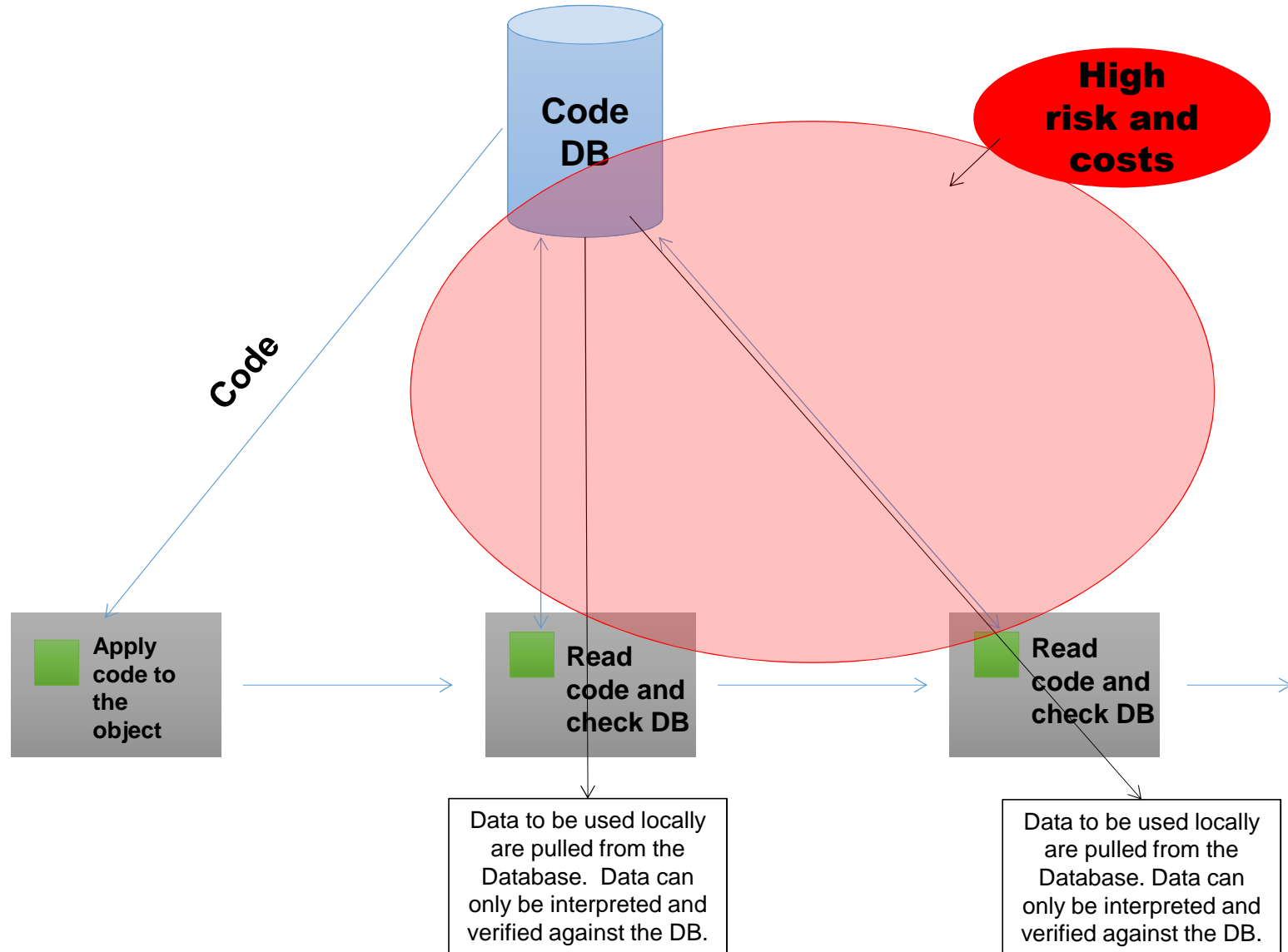
ISO/IEC 20248 provides for interoperable data interpretation, integrity and authority.

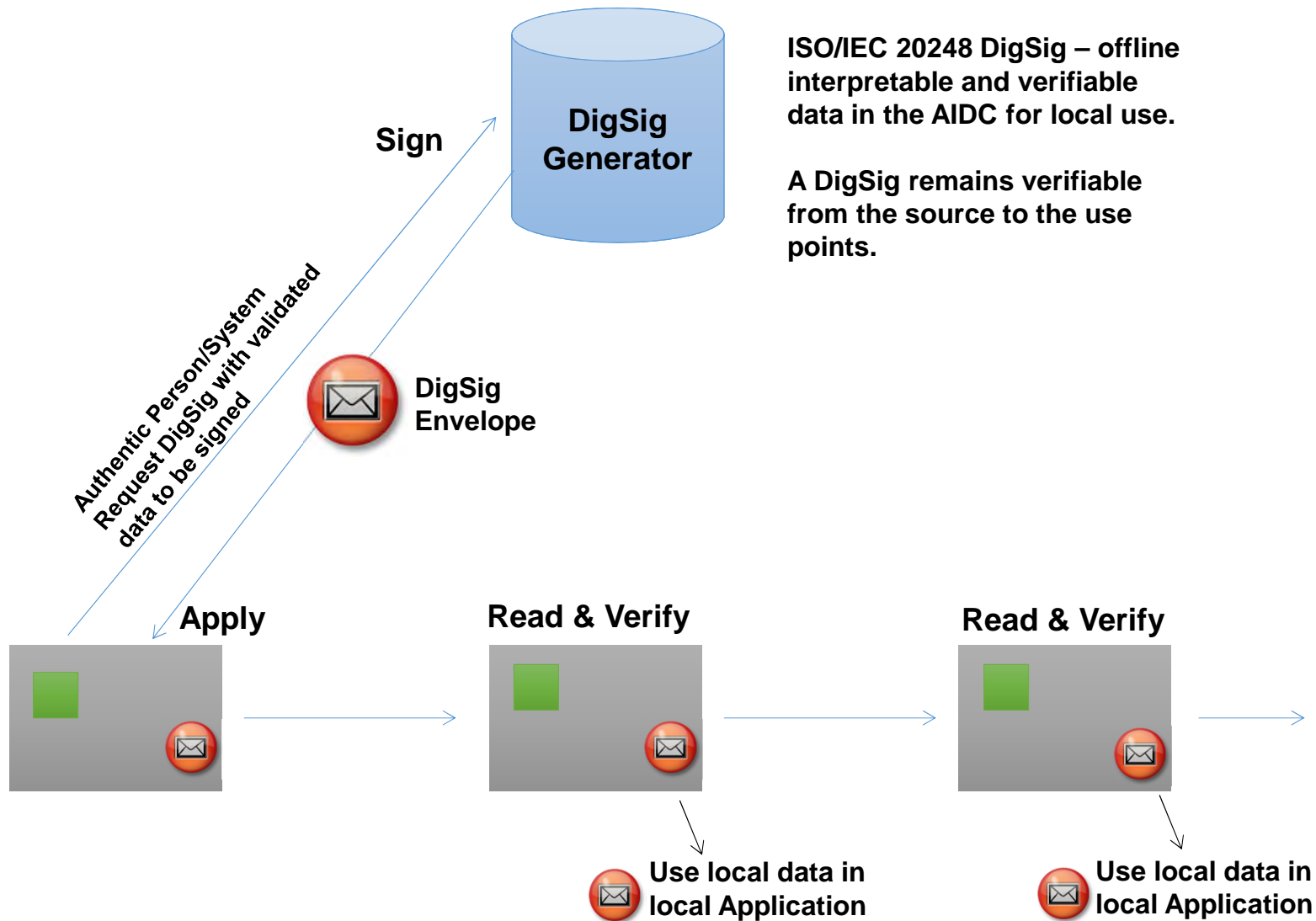
Trust, authority, interpretation and read methods



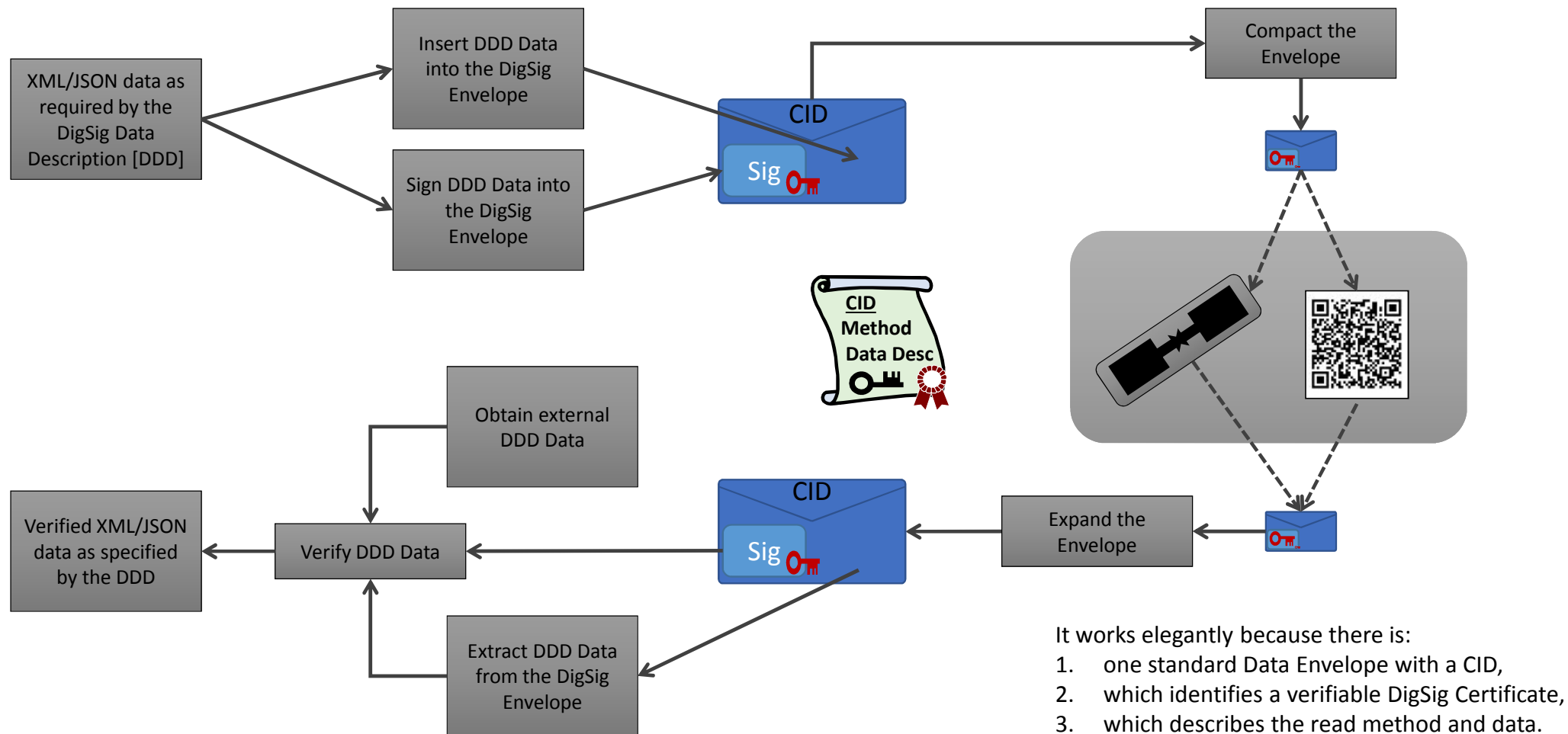
We have successfully shown RAIN enabled systems to work. **We now need to focus on making RAIN enabled systems ROBUST.**







Logical DigSig Data Path



The DigSig Envelope and Data Fields

DigSig{CID, Signature, timestamp, a, b, c} – The CID, Signature and Timestamp are part of the Envelope. The CID is always the first field

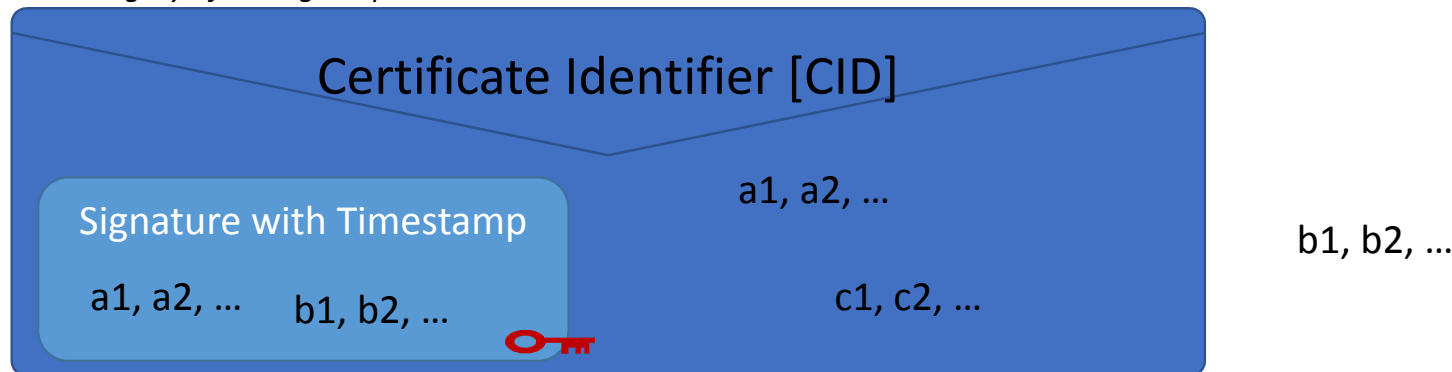
DigSig{a, b, c} – The DigSig envelope as specified by ISO/IEC 20248 without the hidden fields.

a, b and c are sets of fields as described by the DigSig Data Description contained in the CID [Certificate Identifier] referenced DigSig Certificate.

a fields are signed and included in the DigSig envelope. All the information is available to verify when the data structure is read from the AIDC (RAIN, barcode...).

b fields are signed but NOT included in the DigSig envelope. Therefore the value of a b field must be collected by the verifier before verification can be performed. This is useful to link a physical object to an AIDC; eg. for counterfeiting detection.

c fields are NOT signed but included in the DigSig envelope. A c field can therefore NOT be verified, but changed without affecting the integrity of the signed parameters.



DigSigs on 18000-6C/G2V2: A vehicle ID example

UII (excluding PC bits)	TID	User Memory
DigSig{ <u>CID</u> , Reg#, [Ser#], [V ^{vis}], [VIN], [E#], [Expire],	TID,	Signature, Timestamp}

The sequence of read is:

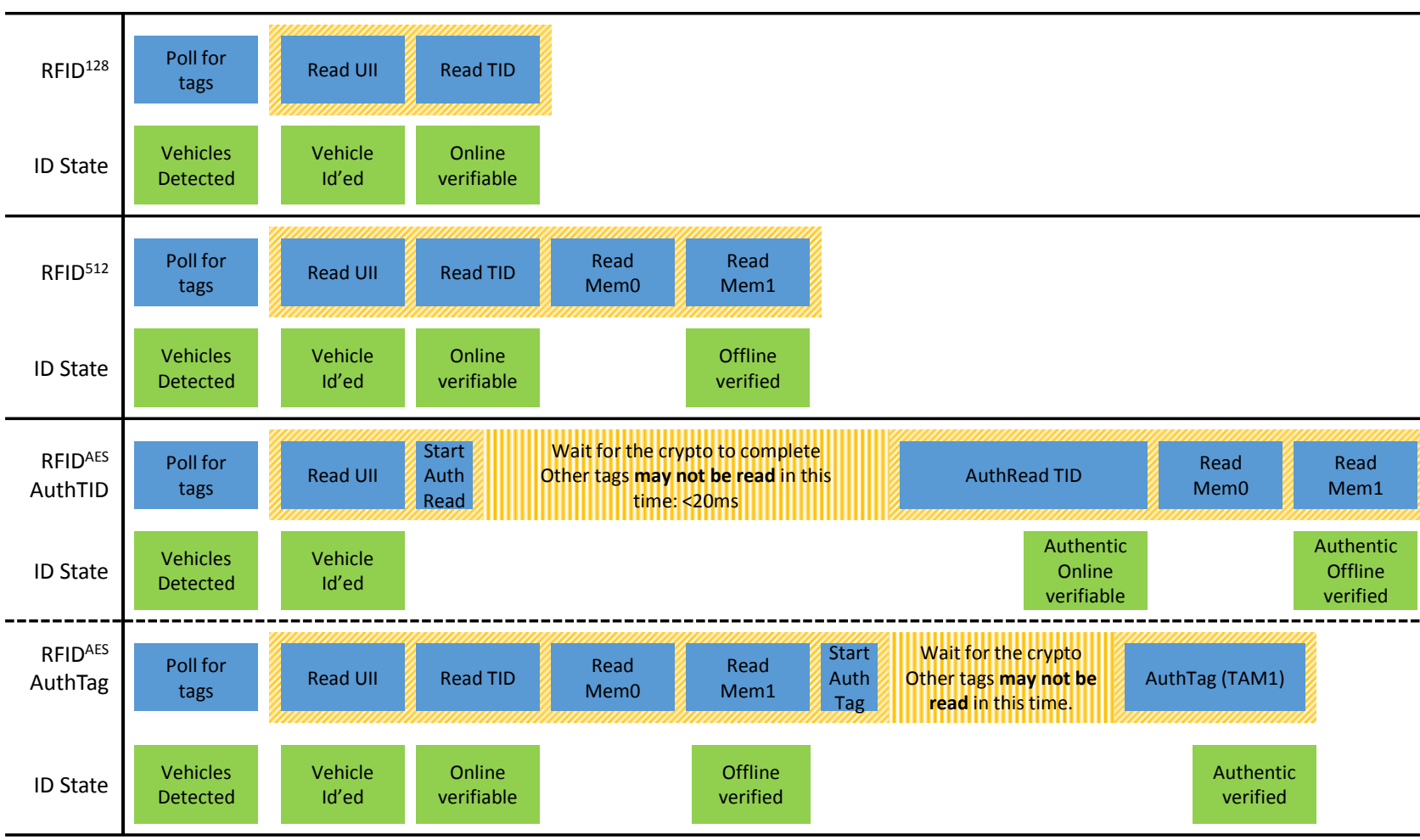
1. UII (during Inventory) providing vehicle identity and information for local decision making, ie. prioritising emergency vehicles,
2. TID providing tag identity for online use and offline copy detection, and
3. User Memory providing the signature and additional information for verification. The vehicle can therefore still be identified even in the bad read performance case.

Note: The optional data, for example [Ser#], [Vvis], [VIN], [E#], [Expire] may also be moved in its DigSig position to place it in User Memory.

Note: CID is always the first field. Signature and timestamp can be placed anywhere in the envelope.



Vehicle ID using RFID^{AES} ISO/IEC 29167-10

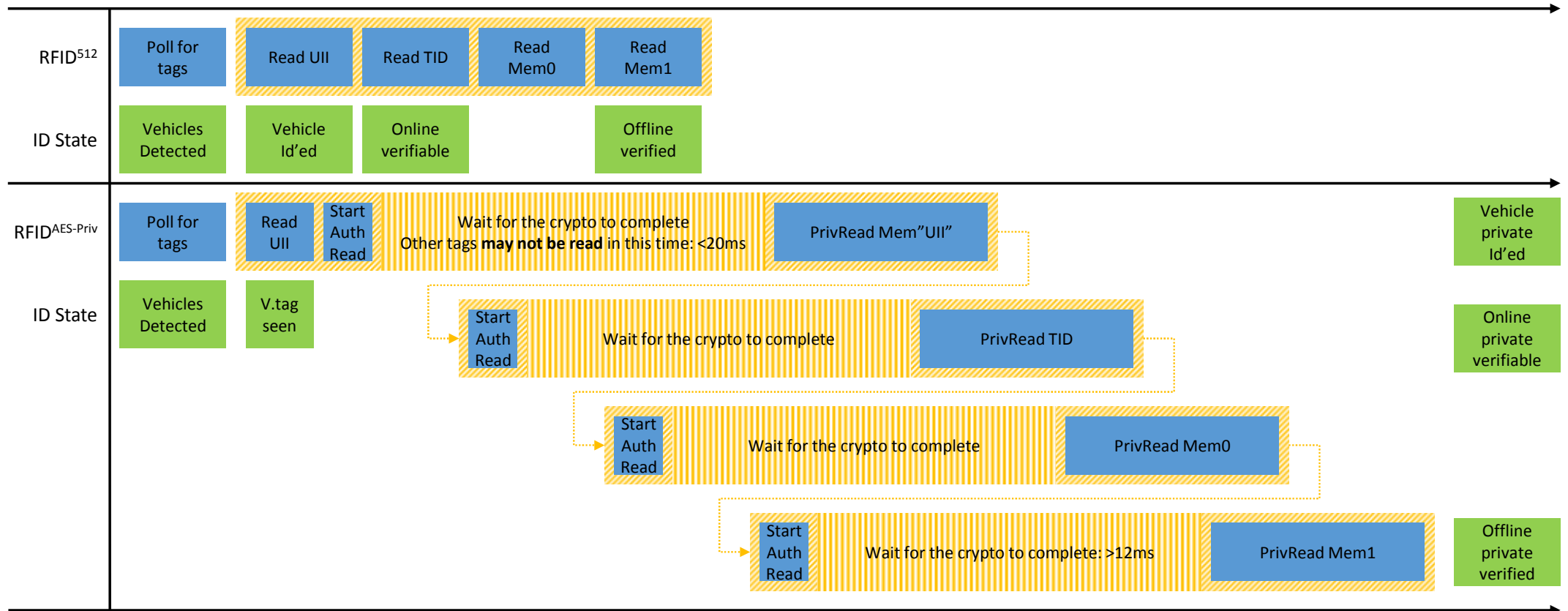


Atomic read
Fail on retry count per
type of read as per
profile



RFID^{AES-Priv} read cycle

RFID^{AES-Priv} read follows the same steps than RFID⁵¹² but the UII and TID is mark untraceable by reducing the visible portion to the CID and chip class identifier. Both fields are reduced to 32 bits. Inventory of the tag therefore only provides the DigSig CID and if the TID is read only the type of chip. The complete DigSig is stored in the User Memory in three 128 blocks: The vehicle identifiers in block 1, and the balance in block 2 and 3.



Examples of a b field



QR, NFC or RAIN to identify a device ownership using a PIN

http://sbox.idoctrust.com/verify/?C=2960&B=EgM6zQsbx4zu-drLK4TlhZBZvIU1CC_wk0WClimolpSilnciQolawg5goEwQ9e0mG3GNQIQYgA=DigSig{PIN, Name, EmployeeIDNumber, AssetSerialNumber}


Note, the above bit count is typically too big for RAIN; the RAW envelope format will be the binary of "EgM6zQsbx4zu-drLK4TlhZBZvIU1CC_wk0WClimolpSilnciQolawg5goEwQ9e0mG3GNQIQYgA="

Freight waybill with a QR referencing a RAIN container seal

DigSig{VehicleLicenseNumber, Source, Destination, SealTID}

Note, the system knows who the sealer is and the DigSig provides the Domain Authority and the time stamp for when the container was sealed.




iDocTrust Offline Verification

Digital Signature Verification Result: Pending
Media type: QR_CODE

Click the check mark for each value that is correct and the X for the values that are incorrect to obtain the verification result.

Domain Authority
<https://www.idoctrust.com/>
6 number PIN
123456

Scan barcode

Name
Albertus Pretorius

Employee ID Number
6107305026087

Asset Serial Number
SAMSUNG1234

Multiple DigSigs on RAIN using DigSigs and **c** fields

UII (excluding PC bits)	TID	User Memory
Primary DigSig{ CID , Primary Fields,	TID,	Signature, Timestamp, MultiDigSigList , [DigSig{...}]

MultiDigSigList is part of the Primary DigSig and specified as follows:

MultiDigSigList |= blocksize, [DigSigContainer, ...]

BlockSize |= Enumeration{1 bit, 16 bits, 32 bits, 128 bits} *encoded as 2 bits providing for the 4 values.*

DigSigContainer |= Start-OffsetBlock, Length-Blocks, Key#

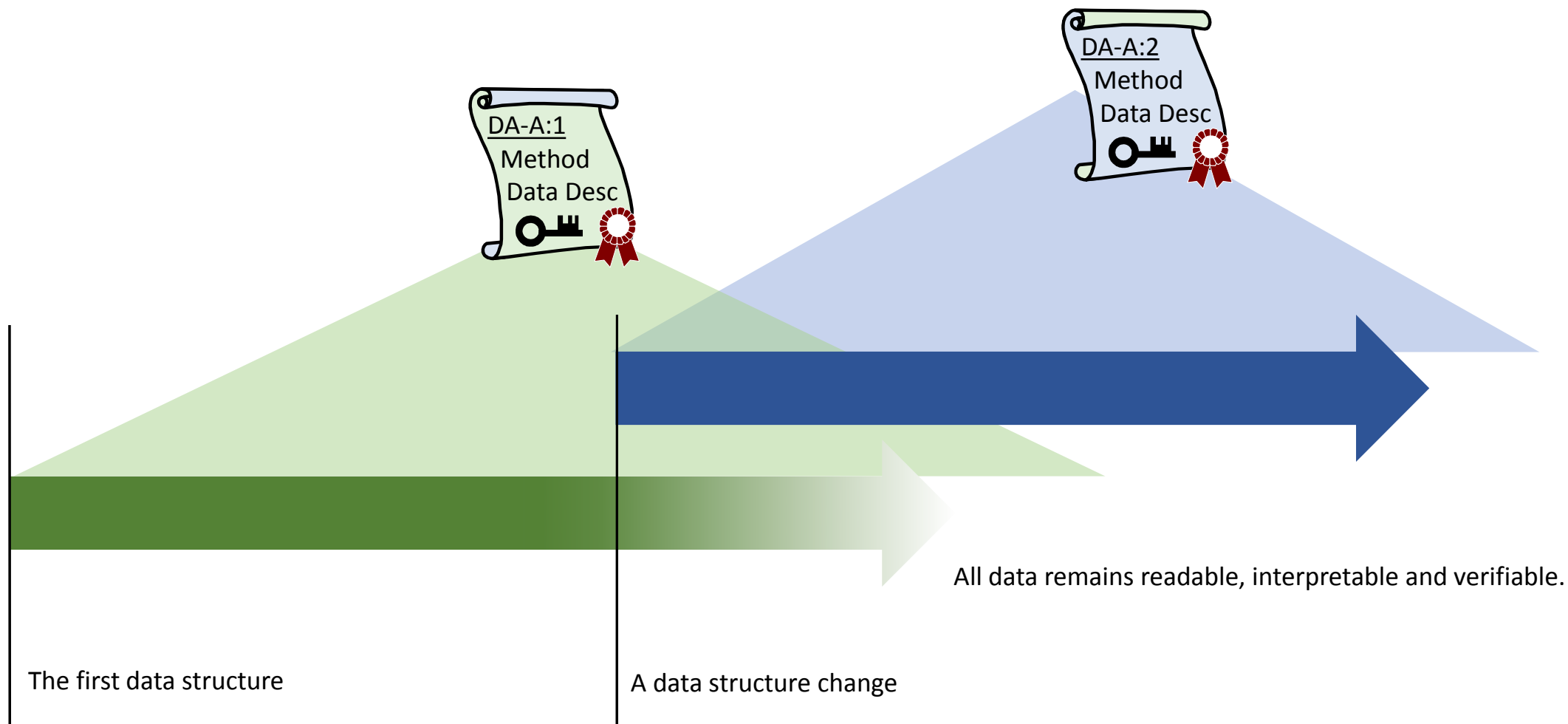
Key# = zero means “not Encrypted” – this example may work for ISO/IEC 29167-10

MultiDigSigList may be change at any time without influencing the Primary DigSig’s integrity.

The Data Description

- Data types: strings (including Kanji), binary fields, numbers and date&time.
- Optimum and dense compaction using:
 - Range
 - Enumeration
 - Character sets
 - Relative time and time grain
- Data structure optimisation is achieved by structures, arrays and NULLability
- Input and output interpretation handles all languages
- Pragmas specifies read and storage methods

Seamless data structure rollover



What is the catch?

- We should have a positive and unique link between the data and the object
 - Use an embedded tag – we need to trust the TID of the tag.
 - Encode a DNA/Fingerprint of the object into the DigSig – not always practical.
- We must be able to detect data copying
 - The DigSig does it, but we need to trust the TID of the tag.
- Some applications need untraceability – the crypto chips does it 😊 but we still need to solve the key management problem.
 - A public key chip goes a long way to solve the problem.
 - NXP UCODE DNA chip with private containers and SAM™ AV2 (hardware keystore) also goes a long way.

We need a trust method or trusted referencing method for the key management.

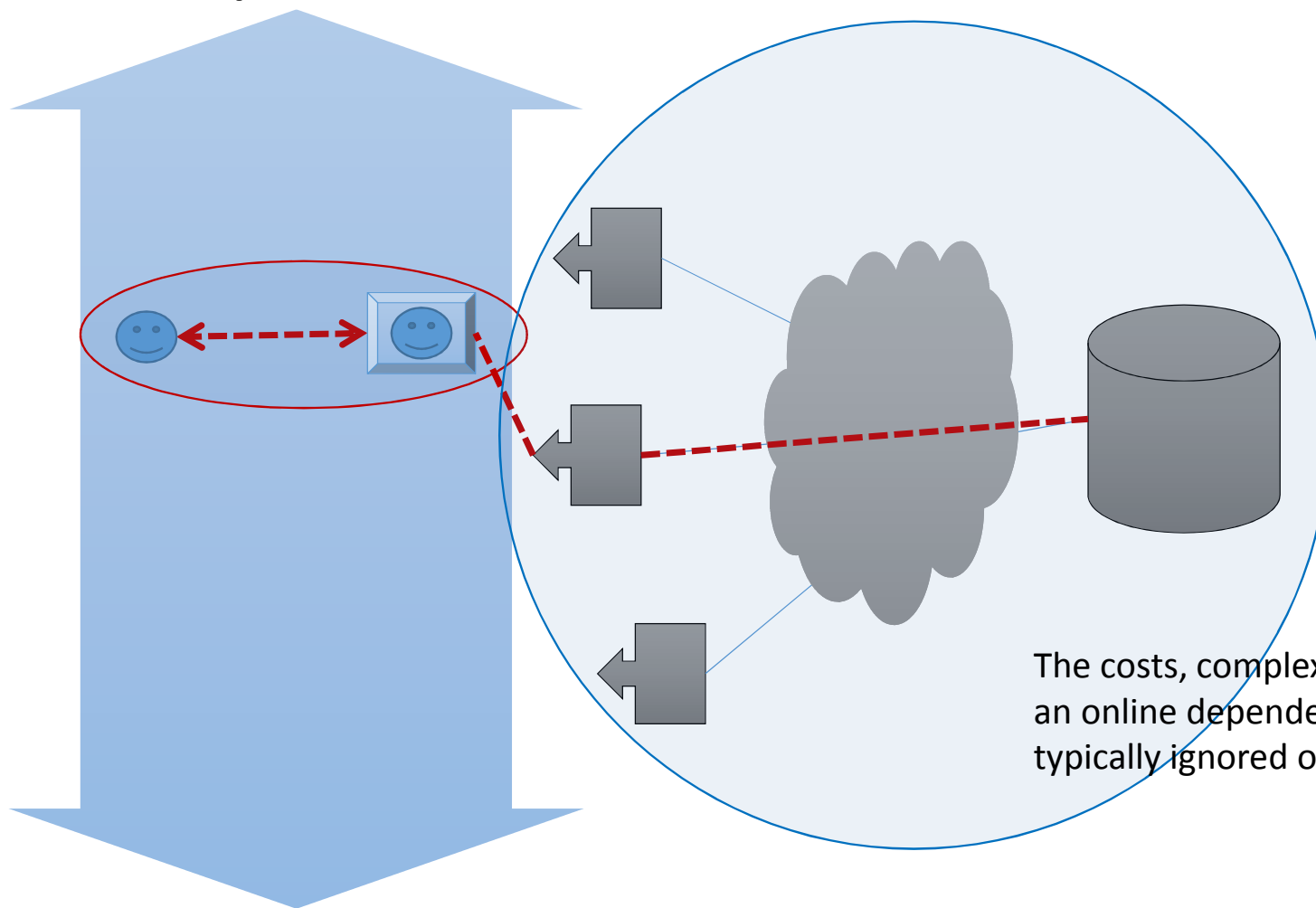
ISO/IEC 20248 may just be the vehicle to simplify key management.



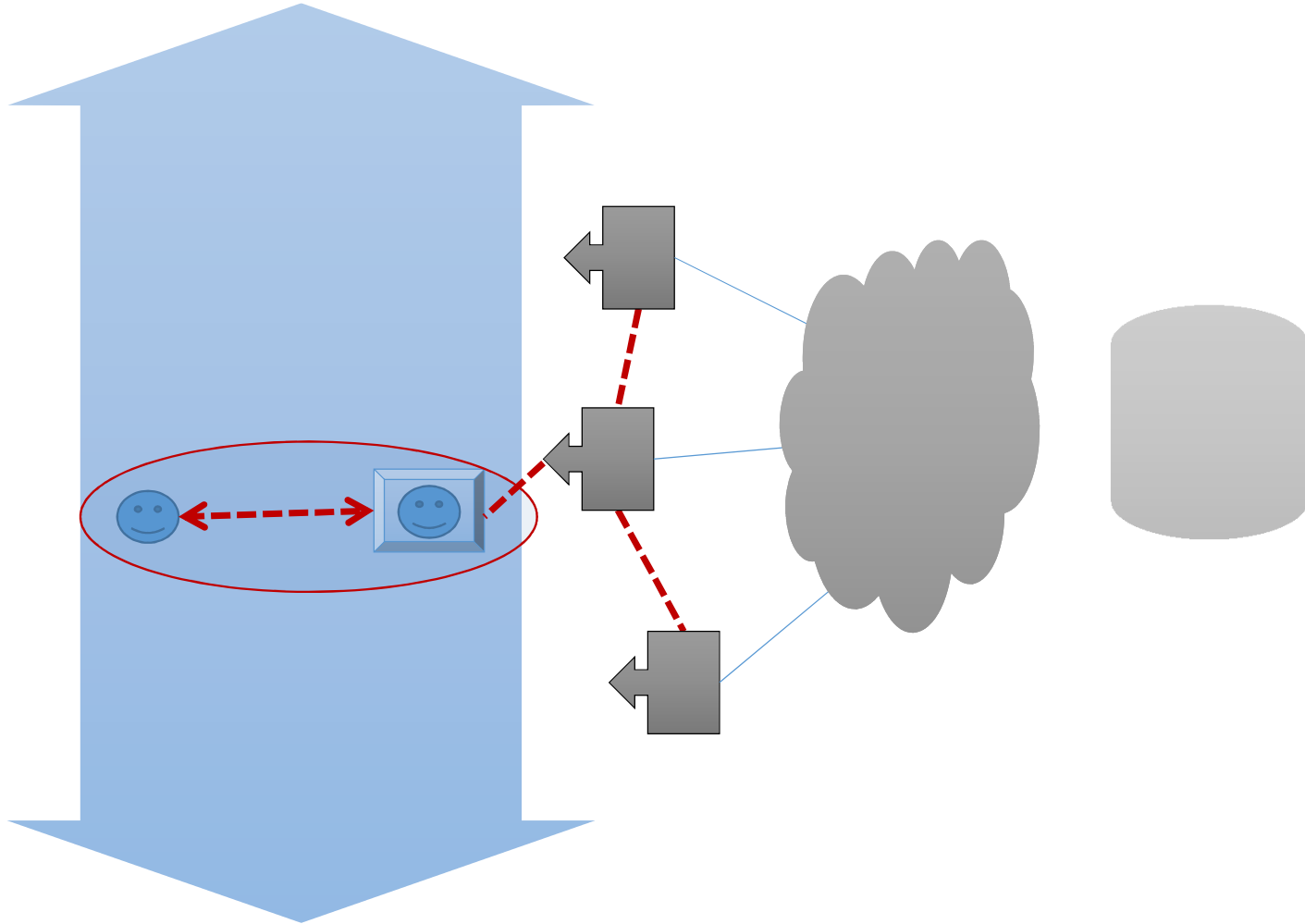
ISO/IEC 20248 (DigSigs) will do for RAIN and IoT what the Browser and HTML did for the Internet

It allows for a secure, open, interoperable and manageable method whereby intelligent agents [and humans] can obtain and exchange verifiable information about real world objects.

Connectivity: Classic ICT and most Cloud methods



Connectivity: Classic IoT



Connectivity: Offline

