# ‘Secure Device Identity’ Profile for TSN-IA

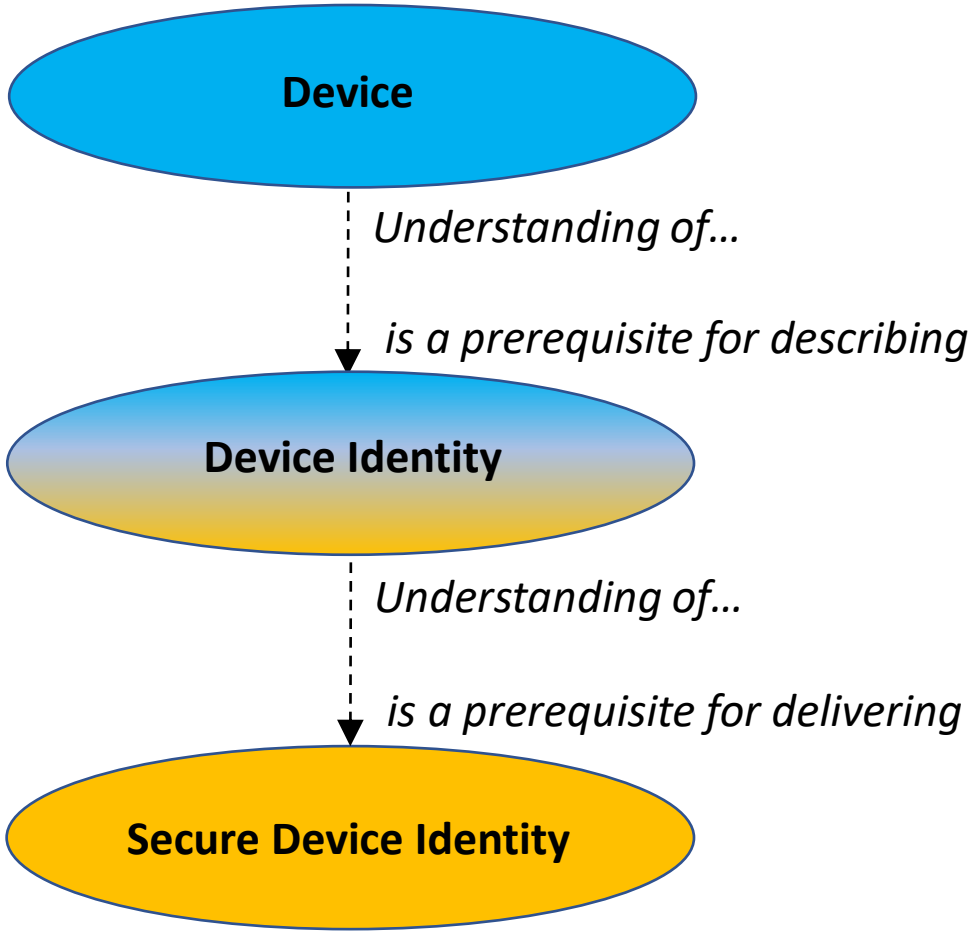
IEEE Interim Session; January 21, 2022

Kai Fischer, Andreas Furch, Oliver Pfaff

# Problem Statement

- Discuss a '**Secure Device Identity**' (IEEE STD 802.1AR-2018) profile for TSN-IA (IEC/IEEE 60802) encompassing:
  - A common **IDevID EE certificate** design for IA-stations: this is meant to be a profile of IEEE STD 802.1AR-2018, chapter 8 '*DevID certificate fields and extensions*'
  - A common **'Supplier information'** template for manufacturers of IA-stations: this is meant to be a (partial) instantiation of IEEE STD 802.1AR-2018, chapter 5.5 '*Supplier information*'
- Motivation:
  - '*Device identity*' addresses use cases such as:
    - *Is this piece of equipment an instance of a distinguished type?*
    - *Was it produced by a specific manufacturer?*
  - A '*secure device identity*' provides safeguards for such use cases. It also allows to protect the setting-up of security

# First Things First




**IA-controller, IA-device, bridge…**

IA-controller, IA-device, bridge…
***manufacturer, order ID, serial number, …***

*Manufacturer, order ID, serial number, …*
***protected by end entity certificates and private keys***

# Background and Rationale

1. The IETF mandates security for NETCONF/YANG (RFC 6241) - *security-always-on*
   - Each-and-every NETCONF/YANG exchange must be protected including the initial one(s)
   - This presents a challenge when IA-stations are in factory default state
2. The Time-Sensitive Networking Profile for Industrial Automation (CD4 resp. D1.3)
   - Uses NETCONF/YANG for network resource management
   - Requires NETCONF-over-TLS (RFC 7589) as a secure transport for NETCONF/YANG
   - Requires IA-stations to have IDevID credentials and trust anchors (IEEE STD 802.1AR) – to be able to protect the initial NETCONF/YANG exchanges i.e. safeguarding the security set-up
3. Caveat: manufacturer-proprietary IDevID EE certificate designs will end in a ‘Secure Device Identity’ plethora
   - Custom IDevID EE certificate designs will make it hard to impossible to fulfill fundamental use cases in industrial automation in an efficient and secure way (see next slides for evidence)

➔ **The ‘Secure Device Identity’ profile for TSN-IA aims at avoiding this Tower-of-Babel**

# IDevID EE Certificate Design Decisions

- Assumption: there is a common ‘Device Identity’ model in a domain comprising information items such as a) ***VendorID***, b) ***DeviceID***, c) ***OrderID***, d) ***SerialNumber***, e) ***HWRevision***, f) ***SWRevision***
- Design decisions that have to be made in order to issue IDevID credentials include (but are not limited to):
  1. *Subset(s)*: which information of a)..f) shall be represented?
  2. *Appearance*: how shall this information appear (e.g. by-value or by-ref)?
  3. *Atoms*: which ASN.1 data types shall be used?
  4. *Structure*: which ASN.1 structures (tagging/composition) shall be used?
  5. *Home*: where to place the resulting structure(s) in an X.509 certificate object?
- Risk: without a common IDevID design, a common ‘Device Identity’ model **will get fragmented** i.e. represented by many independent IDevID certificate designs (see next slide for implications)

# *Antipattern*: Proprietary IDevID EE Certificate Designs

Data sheets from A

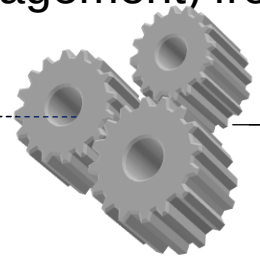
Metadata issued by A AND **structure given by IEC/IEEE 60802**


…

Data sheets from Z

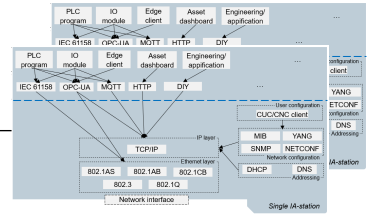
Metadata issued by Z AND **structure given by IEC/IEEE 60802**


Tool (engineering, asset management) from *



***How to automatically (and securely) check isInstanceOf, wasManufacturedBy…?***

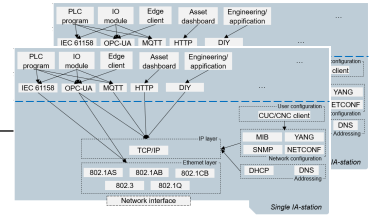
Devices from A



IDevIDs EE cert issued by B AND **according A-custom design**

…

Devices from Z



IDevIDs EE cert issued by Z AND **according Z-custom design**

- *Must comprehend IDevID EE certificate designs by manufacturers A..Z*
- *Must parse IDevID EE certificate objects according A..Z design*
- *Must map IDevID EE certificate values according A..Z value sets*
- *Can not assume to encounter the same set of manufacturers at next occasion (other deployments or later additions)*
- *…*

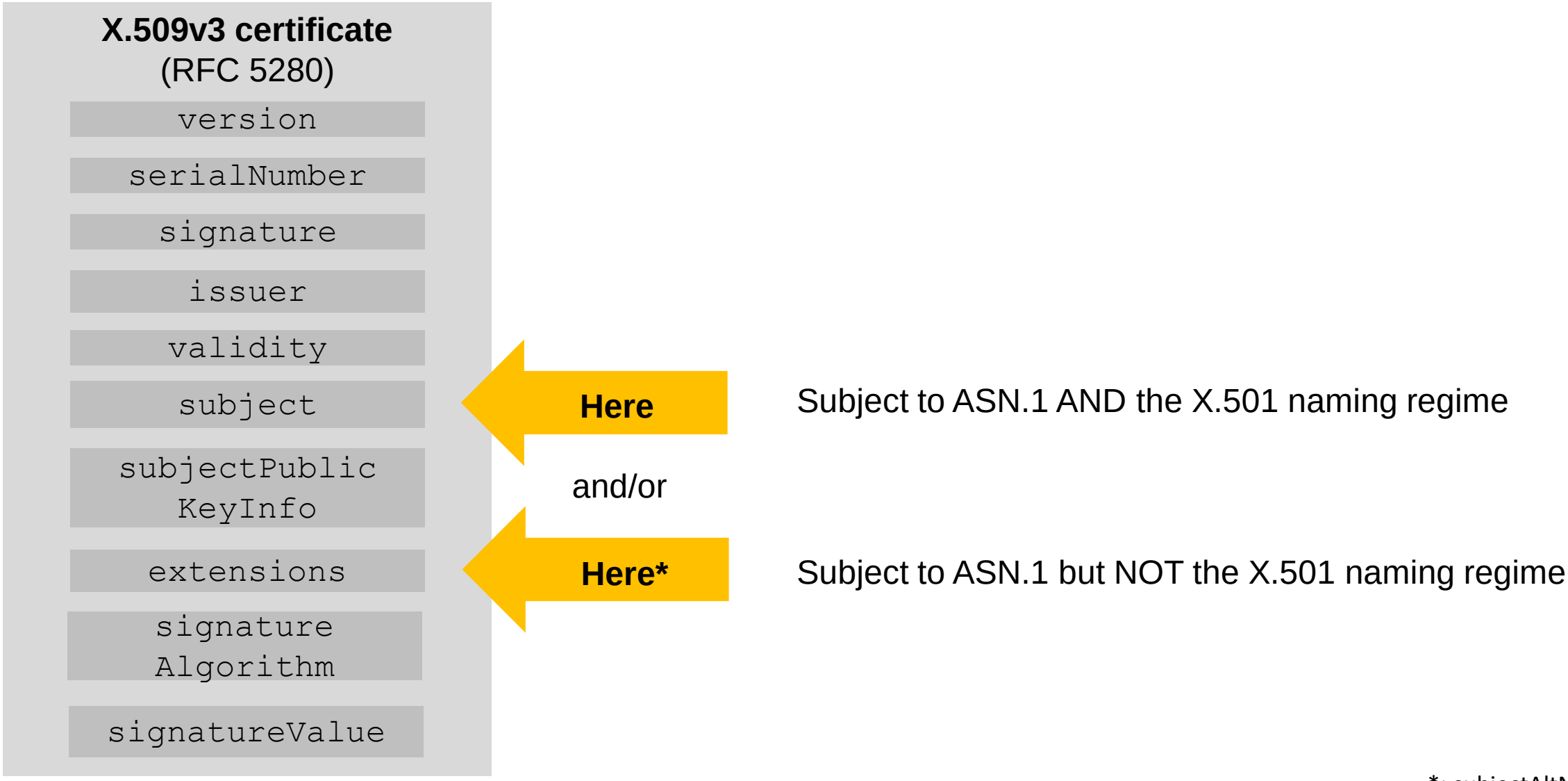
# Key Messages

- A common IDevID EE certificate design, based on IEEE 802.1AR, is needed in industrial automation to solve basic use cases around inventory and setting-up security in an interoperable and efficient way
- The IEC/IEEE 60802 profile for a ‘Secure Device Identity’ aims at a common design in industrial automation. Note:
  - This is in incubation
  - This deck kicks off the discussion

# Hypotheses for Inheriting IDevID EE Certificate Designs

- *Can not fly*: **802.1AR➔manufacturer profile**
  - Expecting industrial use cases e.g. "*is this instance of a physical component member of a specific class*" to work for users that run deployments with products from multiple manufacturers would be naïve
  - See slides above for rationale
- Can fly: **802.1AR➔60802 profile➔middleware/application sub-profile(s)➔manufacturer sub$^2$-profile**
  - The related responsibilities are meant to be:
    i. 802.1AR: IEEE 802.1 Security Task Group
    ii. 60802 profile of i: IEC/IEEE 60802, this slide-deck kicks off a discussion of an IEEE 802.1AR profile for industrial automation
    iii. Middleware/application-specific sub-profile(s) of ii: organizations owning the corresponding specifications e.g. PI, ODVA , OPCF… (at their own discretion)
    iv. Manufacturer sub$^2$-profile of ii or iii: individual manufacturers (at their own discretion)

# Landing Strips for Naming/Properties in IDevIDs




*: subjectAltName

# `Subject` Field (X.501): *No Fit ➔ Ignored*

- Model: information (DIB) is organized in a **single, hierarchical tree** (DIT) with unique nodes identified by DNs comprised of RDNs
  - Can have custom RDNs but overall information are confined to a singular tree, see e.g. chapter 2.3 “Subject Naming in PKIX Certificates” in RFC 6125
- Awkward to impossible to express a **forest-style** set of information items. This is the case in industrial automation:
  1. IEC/IEEE 60802: has an own perception of how to model a HW-based component (IA-station); this ‘*device*’ model should be assumed to come with an own ‘*device identity*’ model
  2. IEC 61158: specific types have their own perceptions of how to model the same component along with an specific information model (the backup slides provide an example for PROFINET)
  3. OPC-UA: has another perception of how to model the same component along with an OPCF information model
  4. DIY: individual manufacturers will have yet another perception of how to model this component along with a DIY information model

# `subjectAltName` Extension: *Fit ➔ Used*

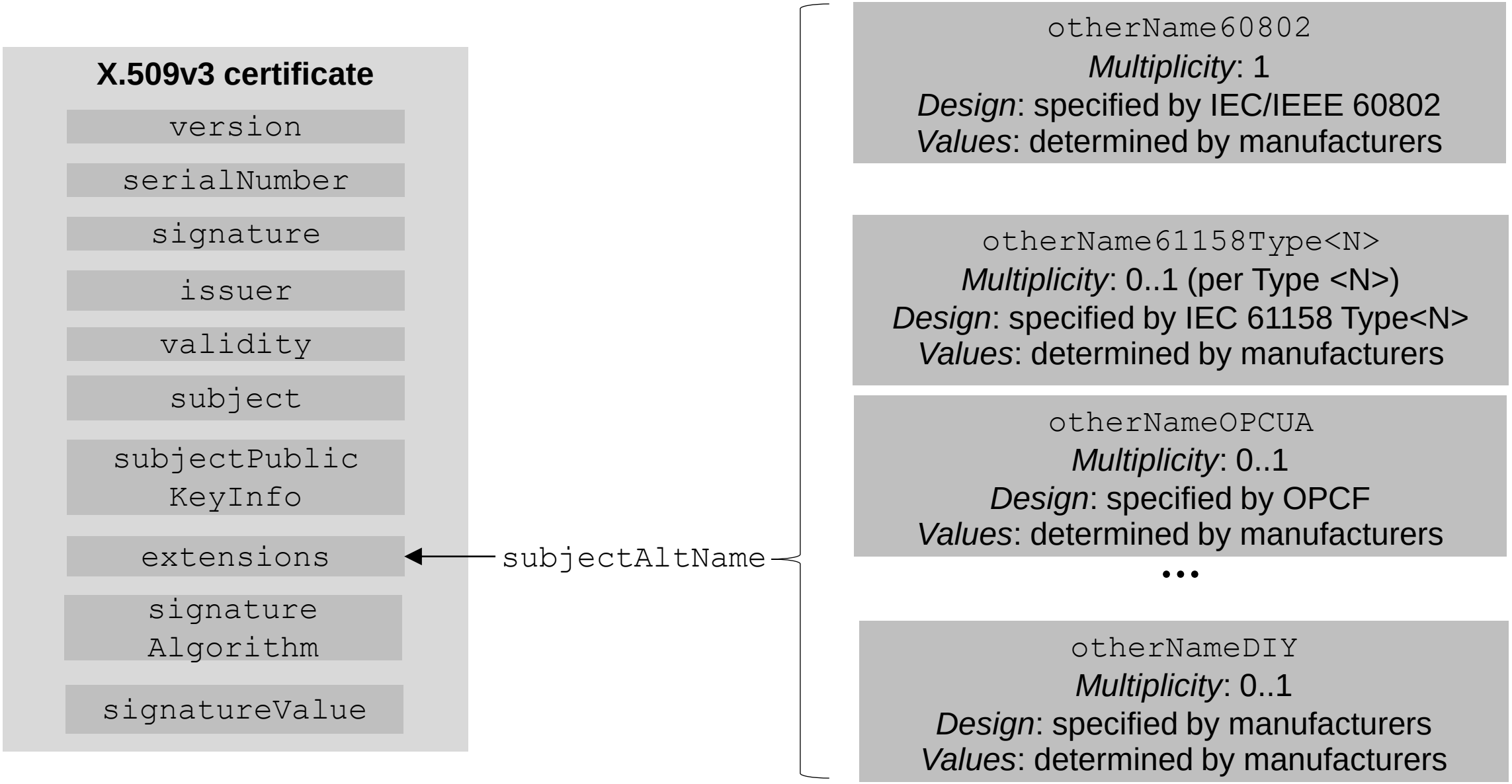
- The `subjectAltName` is an X.509v3 extension that is defined by RFC 5280. It was invented to support naming/property information models that do not fit the X.501 naming regime
- It supports manifold sets of information items about the subject of an X.509 certificate

```
SubjectAltName ::= GeneralNames

GeneralNames ::= SEQUENCE SIZE (1..MAX) OF GeneralName
```

- It has an extension point for a self-coined set of information items about the subject of an X.509 certificate

```
GeneralName ::= CHOICE {
        otherName [0] OtherName,
        … }

OtherName ::= SEQUENCE {
        type-id OBJECT IDENTIFIER,
        value [0] EXPLICIT ANY DEFINED BY type-id }
```

# Modelling Manufacturer Data: Divide&Impera, Tiles



# `OtherName` Standard Incarnation for IEC/IEEE 60802

- `type-id`: selecting an OID value is an IEC/IEEE 60802 responsibility. This can be done by e.g. allocating a value in the OID arc of IEEE 802.1 (`1.0.8802.1`, see https://1.ieee802.org/assigned-numbers/)
- `value`: specifying an ASN.1 structure is an IEC/IEEE 60802 responsibility. As a simple and rough upfront proposal (further elaboration is needed):

```
otherName60802Value = SEQUENCE {

        macAddresses  SEQUENCE SIZE (1..MAX) OF BIT STRING(SIZE(48))

        …}
```

# `OtherName` Standard Incarnations for Middleware/ Applications

- Decisions and actions are a duty of the specification owner resp. the manufacturers of components for this domain. This includes:
  - Specification owner responsibilities e.g. PI (PROFINET), ODVA, OPCF or …:
    - *Decisions*: whether to specify a ‘Secure Device Identity’ sub-profile under the authority of this specification. Whether to make its support optional or mandatory.
    - *Actions* (if needed): specify the structure for an `OtherName` incarnation that allows to express the manufacturer data according the ‘Device Identity’ information model of this domain
  - Manufacturer responsibilities:
    - *Decisions*: whether to utilize a ‘Secure Device Identity’ sub-profile for this domain (if specified and optional)
    - *Actions* (if needed): establish manufacturer data values, create `OtherName` objects for this domain-of-interpretation, issue corresponding IDevID EE certificate(s), create corresponding IDevID credential(s) and build them into product instances

# `OtherName` Custom Incarnations for Manufacturers

- All related decisions and actions remain an individual duty of an individual manufacturer. This includes:
  - *Decisions*: whether to specify and utilize a ‘Secure Device Identity’ sub[2]profile for own purposes
  - *Actions* (if needed):
    - Specify the structure for an `OtherName` DIY incarnation that allows to express the additional manufacturer data in the information model for its products
    - Instance-level actions: establish the manufacturer data values, create `OtherName` DIY objects for this domain-of-interpretation, issue corresponding IDevID EE certificate(s), create corresponding IDevID credential(s) and build them into product instances

# Illustration for IDevID EE Cert Object Heritage

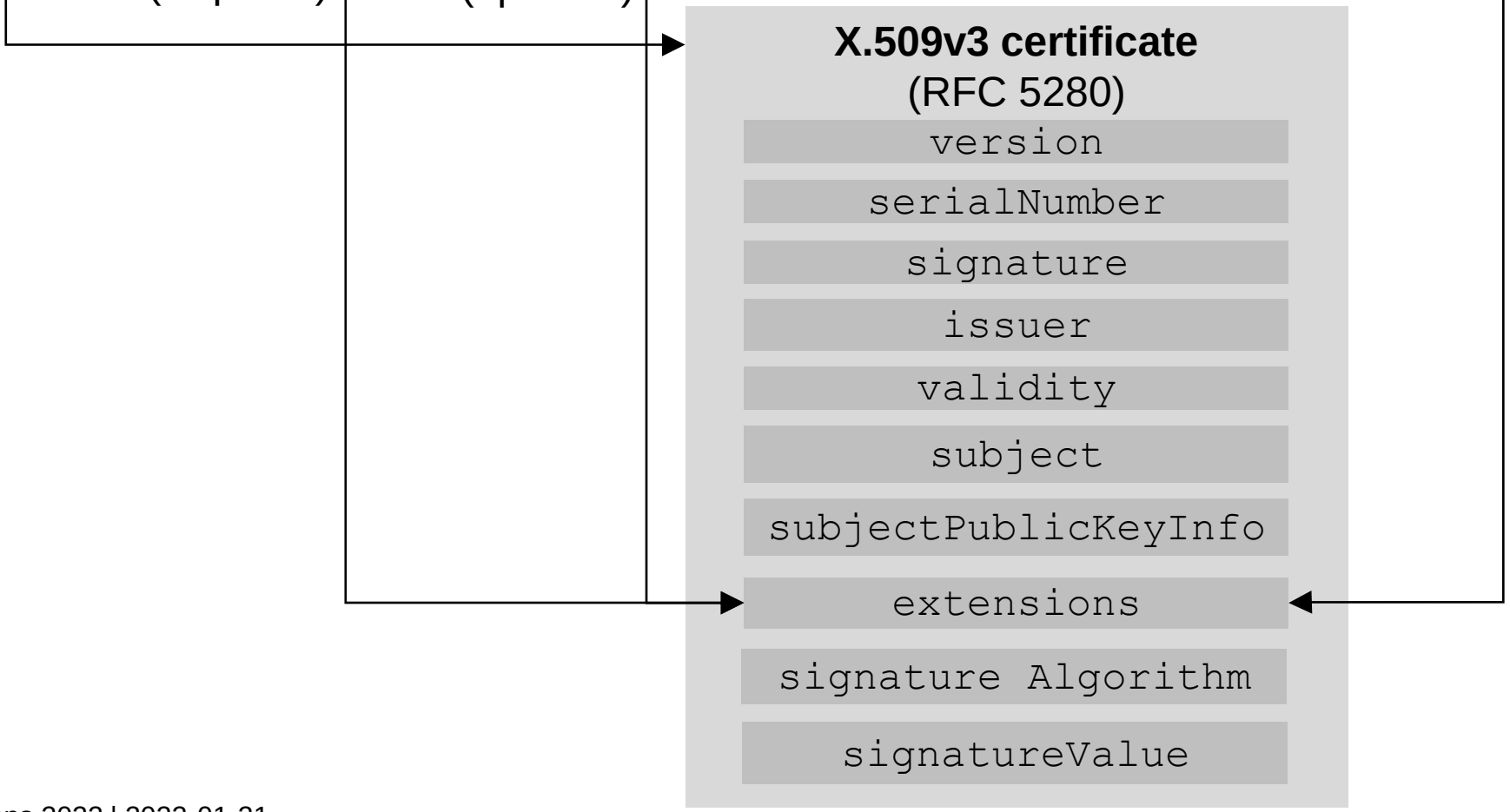
**802.1AR➔60802 profile➔middleware/application sub-profile(s)➔manufacturer sub²-profile**

Basis for occupying fields/contents in X.509 certificates for physical devices

`OtherName(s)` for IEC/IEEE 60802 (required)

`OtherName`(s) for other specs (optional)

`OtherName` for manufacturer DIY (optional)



# IDevID Credential Multiplicity

- Default case: #1 physical product resp. orderable item = #1 IDevID credential with
  - #1 IDevID credential = #1 IDevID EE certificate
  - #1 IDevID EE certificate = #1..n `otherName` incarnations
    - The `otherName60802` object is issued by the manufacturer according the design that is specified by IEC/IEEE 60802
    - Zero, one or more `otherName<..>` objects issued by the manufacturer according the design that is specified by the <…> specification owner e.g. PI for PROFINET
    - Zero or one `otherNameDIY` objects issued by the manufacturer according its own design
- Special case: #1 physical product resp. orderable item = #n IDevID credential to cover
  - Multiple cryptographic algorithms or key lengths (`subjectPublicKey` and/or `signature`)

# Follow-Ups Are Needed

- ‘Device identity’ model for IA-stations (IEC/IEEE 60802)
- IEEE STD 802.1AR, chapter 5.3 ‘Required capabilities’ and 5.4 ‘Optional capabilities’
  - Requirements on IDevID EE/CA certificates beyond the `subject` field resp. `subjectAltName` extension in IDevID EE certificates
  - Requirements beyond IDevID EE/CA certificate objects e.g. internal/external private key generation
- IEEE STD 802.1AR, chapter 5.5 ‘Supplier information’
  - Requirements beyond the ‘device’ component e.g. obligation to publish objects (certificates, CRLs, policies) or provide services (means to request revocation status)

# Abbreviations

| | |
|---|---|
| ASN.1 | Abstract Syntax Notation no. 1 |
| Cert | Certificate |
| CNC | Centralized Network Configuration |
| CRL | Certificate Revocation List |
| CUC | Centralized User Configuration |
| DAP | Device Access Point |
| DHCP | Dynamic Host Configuration Protocol |
| DIT | Directory Information Base |
| DIT | Directory Information Tree |
| DIY | Do It Yourself |
| DN | Distinguished Name |
| DNS | Domain Name Service |
| GSD | General Station Description |
| HTTP | HyperText Transfer Protocol |
| I&M | Identification and Maintenance |
| IA | Industrial Automation |
| ID | Identifier |
| IO | Input Output |
| IOC | IO Controller |
| IOD | IO Device |
| IP | Internet Protocol |
| MAC | Media Access Control |
| MIB | Message Information Base |
| MQTT | Message Queuing Telemetry Transport |
| OID | Object ID |
| OPC | Open Platform Communications |
| OPCF | OPC Foundation |
| PI | PROFINET International |
| PLC | Programmable Logic Controller |
| PROFINET | PROcess FIeld NETwork |
| RDN | Relative DN |
| SNMP | Simple Network Management Protocol |
| STD | STandarD |
| TCP | Transmission Control Protocol |
| TLS | Transport Layer Security |
| TSN | Time-Sensitive Networking |
| UA | Unified Architecture |
| VIN | Vehicle Identification Number |
| YANG | Yet Another Next Generation |

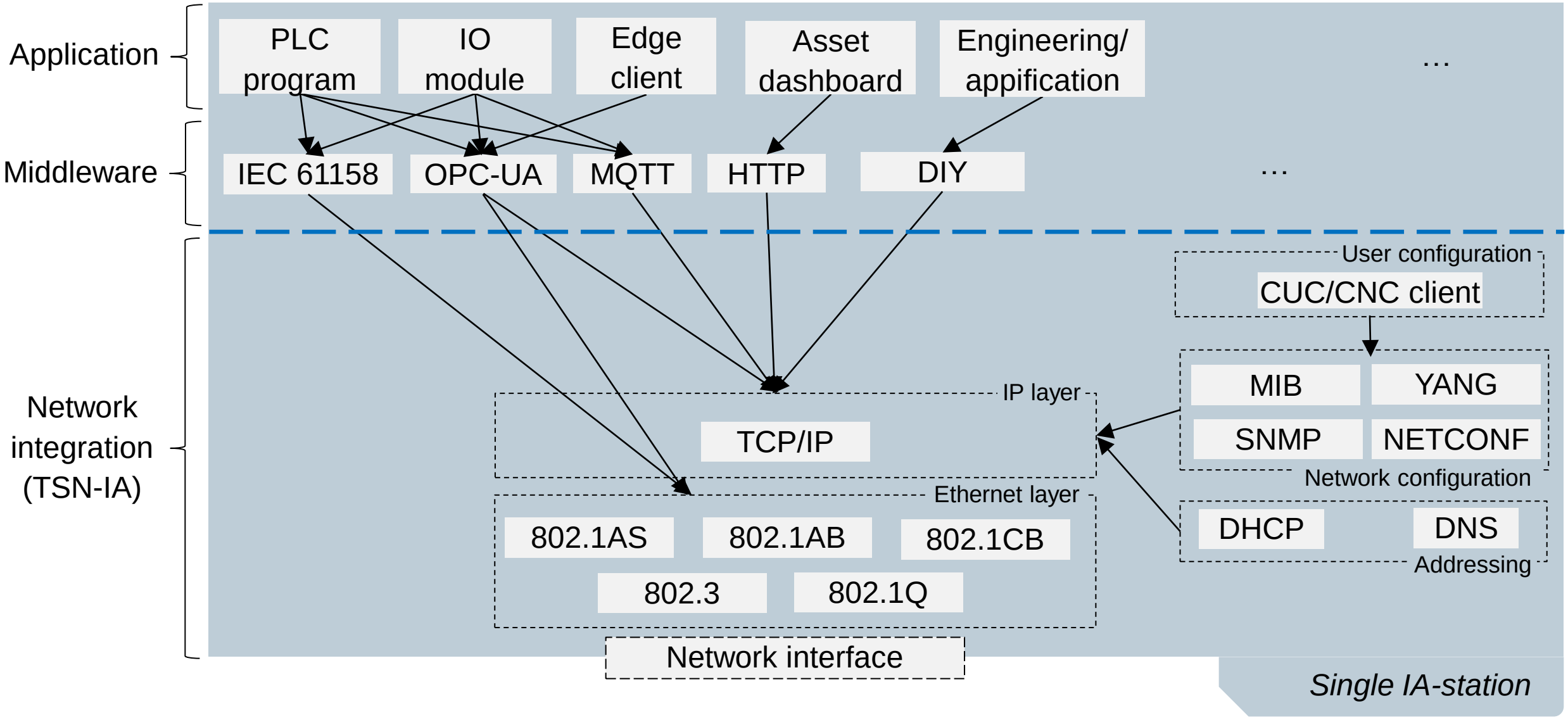
# Contacts

Kai Fischer, Siemens AG, T CST SES-DE, kai.fischer@siemens.com

Andreas Furch, Siemens AG, T CST SES-DE, andreas.furch@siemens.com

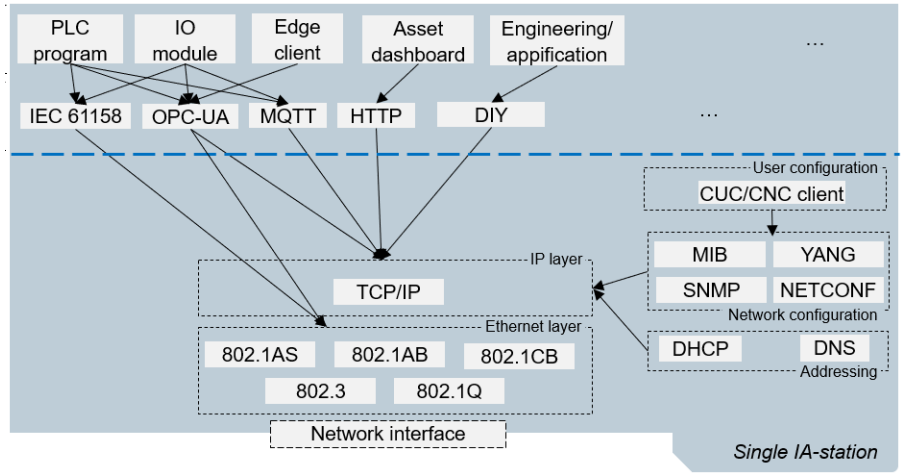
Oliver Pfaff, Siemens AG, DI FA CTR ICO PO, oliver.pfaff@siemens.com

# Ingredients of IA-Stations

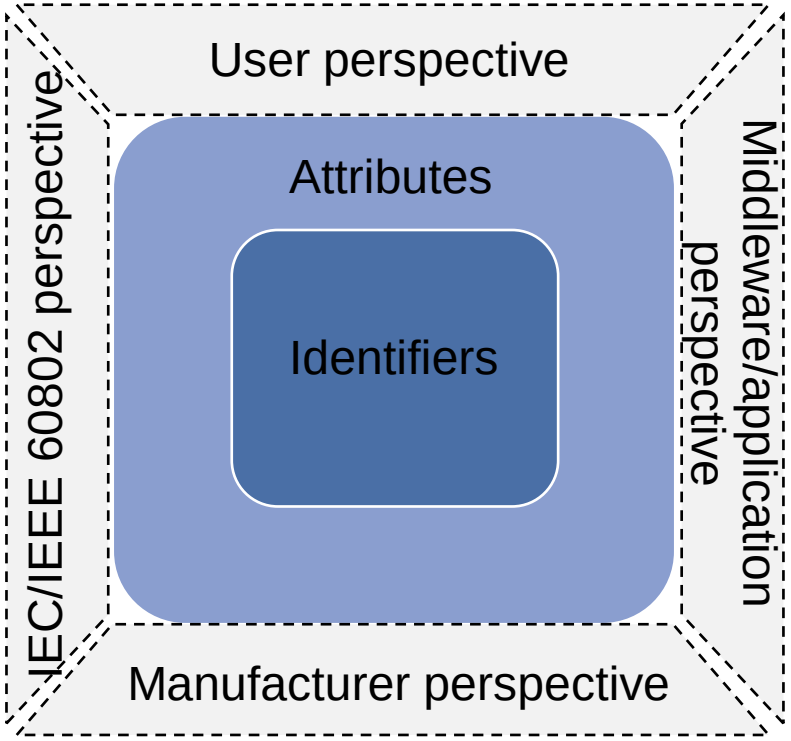


# ‘Device Identity’ of IA-Stations (1)

## ‘Device’ component model



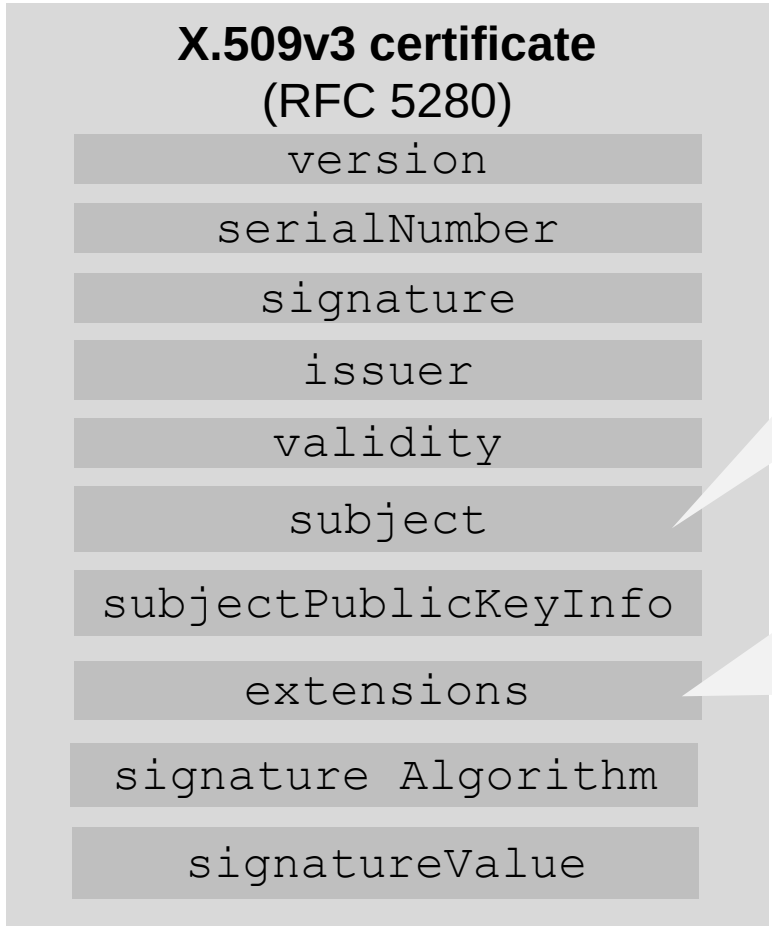

## ‘Device identity’ information model



# ‘Device Identity’ of IA-Stations (2)

- Device identity is the representation of an industrial automation entity (e.g. IA-station) in interactions with other entities or system components
- Device identity comprises unique identifiers and supplementary attributes/properties:
  - Identifiers uniquely refer to an entity - within an understood domain of interpretation – e.g. product serial numbers (example from automotive: VINs such as 5J8TB4H38FL002262)
  - Attributes/properties describe distinct characteristics (non-unique) of an entity – e.g. product class (example from automotive: brand name/model name/model type such as “Volkswagen”/”Golf”/ ”Convertible”)
- The purpose of such device identity items is the fulfillment of use cases e.g. *device replacement without engineering*
- Different stake-holders (IEC/IEEE 60802, middleware/application specification initiatives, component manufacturers, component users) will have different perspectives of the set of identifiers/attributes they are interested in and that are processed by them

# Subject and SubjectAltName According RFC 5280




**Subject** (field): *identifies the entity associated with the public key stored in the subject public key field*

**SubjectAltName** (extension): *allows identities to be bound to the subject of the certificate:*

*i. These identities may be included in addition to or in place of the identity in the **subject** field of the certificate*

*ii. Defined options include an Internet electronic mail address, a DNS name, an IP address, and a Uniform Resource Identifier (URI)*

*iii. Other options exist, including completely local definitions*

*iv. Multiple name forms, and multiple instances of each name form, MAY be included*

*v. Whenever such identities are to be bound into a certificate, the subject alternative name extension MUST be used*

# Use Cases for PROFINET ‘Device Identity’

- ***Engineering**:* VendorID, DeviceID, ModuleIdentNumber, SubmoduleIdentNumber
  - Note: the (VendorID, DeviceID)-tuple identifies a GSD file which provides the data sheet for dealing with a device during its engineering
- ***Device/module/submodule replacement***: VendorID, OrderID of the concerned item
  - The (VendorID, OrderID)-tuple identifies device/module/submodule replacement candidates for the use case ‘Device replacement without engineering’
- ***Device/module/submodule repair/warranty***: VendorID, OrderID, IM_Serial_Number of the concerned item
  - The (VendorID, OrderID, IM_Serial_Number)-triplet identifies a device/module/submodule instance
- ***Device/module/submodule (firmware/software) update**:* VendorID, OrderID, IM_Hardware_Revision, IM_Software_Revision of the concerned item

# Native PROFINET ‘Device Identity’ (1)

- Manufacturer data describing the core ‘Device Identity’ of IO devices/controllers:
  - ***VendorID*** (Unsigned16): PI-assigned manufacturer identifier, see PI catalogue
  - ***DeviceID*** (Unsigned16): manufacturer-assigned type identifier for the product class of an IO device
  - ***OrderID*** (UnicodeString8[64]): manufacturer-assigned class identifier for mutually replaceable product instances, resolution is manufacturer-specific (opaque for owners/operators)
  - ***IM_Serial_Number*** (VisibleString[16]): manufacturer-provided instance identifier, establishes uniqueness in addition to the n-tuple (VendorID, DeviceID, OrderID, IM_Hardware_Revision)*
  - ***IM_Hardware_Revision*** (Unsigned16): manufacturer-provided information on component hardware
  - ***IM_Software_Revision*** (Unsigned8[3]): manufacturer-provided information on component software

*: IM_Software_Revision may change over time due to firmware updates

# Native PROFINET ‘Device Identity’ (2)

- Additional manufacturer data describing the ‘Device Identity’ of IO (sub)modules (which may have their own *VendorID*, *DeviceID*, *OrderID, IM_Serial_Number, IM_Hardware_Revision, IM_Software_Revision)*:
  - ***ModuleIdentNumber*** (Unsigned32): manufacturer-provided type-identifier for IO modules, identifies an IO module type uniquely in addition to the (VendorID, DeviceID)-tuple
    - Note: the ModuleIdentNumber for the DAP module can also serve as DeviceID (in case of media redundancy there can be 2 DAP modules in 1 IO device which have the same ModuleIdentNumber)
  - ***SubmoduleIdentNumber*** (Unsigned32): manufacturer-provided type-identifier for IO submodules, identifies an IO submodule type uniquely in addition to a ModuleIdentNumber
- Owner/operator data items (out-of-scope with respect to an IDevID EE certificate sub-profile):
  - N.a. (such items exist, e.g. I&M1/2/3 objects but are not considered herein)