

Translation tables / rules for IA-stations

Requirements for Plug & Produce

03/22 v01

Make network configured traffic types and PTP instances
available for middleware / application alignment

Thanks to Lukas Wuesteney and Josef Dorr for their support!

Günter Steindl

Siemens AG

Splitting network configuration from middleware/application configuration

Plug & Produce requires the splitting of responsibilities between middleware/application and network provisioning.

This make it impossible for the middleware/application configuration tool to assign PCP/VID values to middleware defined traffic types.

Another topic is the alignment of PTP instance IDs.

Thus, a translation

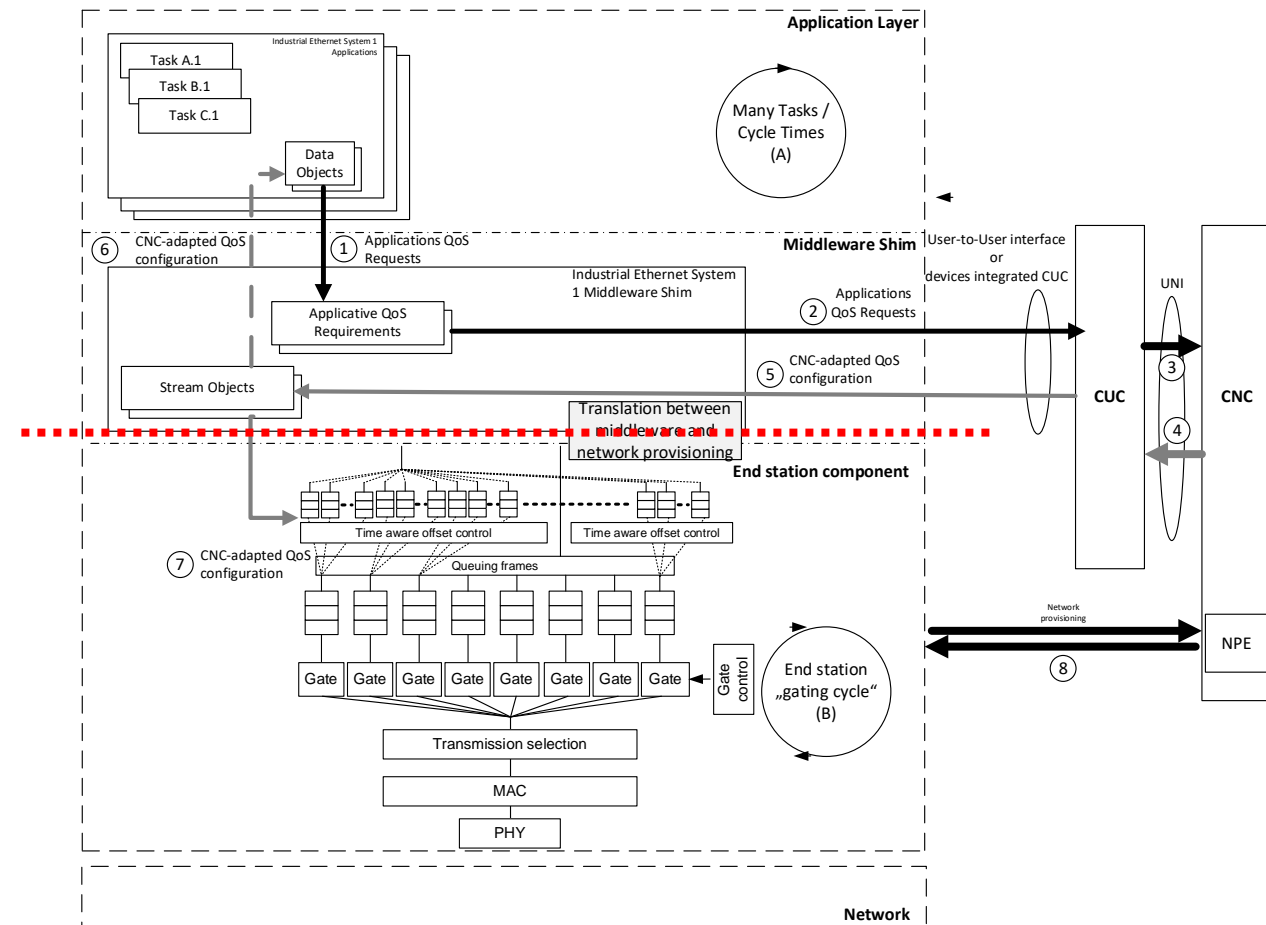
- between middleware defined traffic types **and** network provisioned traffic types is needed.
- between middleware defined ClockTargets and/or ClockSources, **and** network provisioned PTP instance IDs is needed.

Decoupling between middleware and communication

Splitting of responsibilities between middleware and network provisioning requires a standardized way for the middleware to access the information.

The local YANG database contains all network provisioned information.

Thus, a local access for the middleware to this information is required to implement the decoupling.



Traffic Type Translation

Application

Middleware

IEC/IEEE 60802

Application

Application

Middleware A

Middleware n

Middleware translates its Traffic Types into the network provided traffic types!

Middleware Traffic type	IEC/IEEE 60802	Other profile e.g. DetNET or 5G, ...
HIGH	Iso	...
LOW	Cyc	...
RT	ACyc	...
NC	Network Control	...
EV	Event and Alarms	...
CM	Connection Management	...
BEH	BEH	...
BEL	BEL	...

Middleware Traffic type	IEC/IEEE 60802	Other profile e.g. DetNET or 5G
Critical	Iso	...
Medium	Cyc	...
Low	ACyc	...
NC	Network Control	...
EV	Event and Alarms	...
CM	Connection Management	...
BEH	BEH	...
BEL	BEL	...

Traffic type	PCP	VID	VID (red)
Iso	6	101	102
Cyc	5	103	104
ACyc	4	100	---
Network Control	7	---	---
Event and Alarms	3	100	---
Connection Management	2	100	---
BEH	1	100	---
BEL	0	100	---

Provide this information to the different middlewares to support the translation / use of the network configured values.

This values are defined by the CNC and provided by Network Provisioning (NPE).

ifStackTable

VLAN Interfaces in IEEE 802.1Q-2018

Decoupling is nothing new for IEEE 802.1Q – its already defined for MAC and VLAN interfaces.

Principle:

End station component MAC interface hosts multiple VLAN interfaces

⁶ 17.3.4 Relationship of the IEEE8021-Q-BRIDGE-MIB to other MIB modules

⁷ 17.3.4.1 Relationship to the IF-MIB

⁸ This standard assumes the interpretation of the Interfaces Subtree to be in accordance with the IF-MIB
⁹ [RFC2863], which states that the interfaces table (ifTable) contains information on the managed resource's
¹⁰ interfaces and that each sublayer below the internetwork layer of a network interface is considered an
¹¹ interface.

¹² This standard does not make any assumption that within an entity, VLANs that are instantiated as an entry in
¹³ dot1qVlanCurrentTable—either by management configuration through dot1qVlanStaticTable or by dynamic
¹⁴ means (e.g., through MVRP)—are also represented by an entry in ifTable.

¹⁵ Where an entity contains higher-layer protocol entities (e.g., IP-layer interfaces that transmit and receive
¹⁶ traffic to/from a VLAN), these should be represented by an interface that represents the protocol entity and
¹⁷ an interface of ifType l2vlan (135) with the ifStackTable indicating the stacking relationship between the
¹⁸ two entities.

¹⁹ 17.3.4.1.1 ifStackTable

²⁰ In addition, the IF-MIB [RFC2863] defines a table 'ifStackTable' for describing the relationship between
²¹ logical interfaces within an entity. It is anticipated that implementors will use this table to describe the
²² binding of (for example) Link Aggregation Group (LAG) interfaces to physical Ports, although the presence
²³ of VLANs makes the representation less than perfect for showing connectivity. The ifStackTable cannot
²⁴ represent the full capability of this standard, since this standard makes a distinction between VLAN bindings
²⁵ on *ingress* to and *egress* from a Port. These relationships may or may not be symmetrical; whereas, Interface
²⁶ MIB Evolution assumes a symmetrical binding for transmit and receive. This makes it necessary to define
²⁷ other manageable objects for configuring which Ports are in the member set for which VLANs.

VLAN Interfaces in the IETF Interface-list

How could a translation based on the IETF interface entries look like?

Content of name and/or description string shall be profiled by the TSN-IA profile.

Example:

Name := vlan-id-100

Description :=

- List of allowed traffic types and their PCP (optional and their DSCP) value
- Optional vlan-id of the VLAN used for redundancy in case of end station FRER
- [Version 2]
Optional assigned CUC in case of multiple CUC at one host

```
module: ietf-interfaces
  +--rw interfaces
    +--rw interface* [name]
      +--rw name string
      +--rw description? string
      +--rw type identityref
      +--rw enabled? boolean
      +--rw link-up-down-trap-enable? enumeration {if-mib}?
      +--ro admin-status enumeration {if-mib}?
      +--ro oper-status enumeration
      +--ro last-change? yang:date-and-time
      +--ro if-index int32 {if-mib}?
      +--ro phys-address? yang:phys-address
      +--ro higher-layer-if* interface-ref
      +--ro lower-layer-if* interface-ref
      +--ro speed? yang:gauge64
      +--ro statistics
```

Sanity check

How could a translation based on the IETF interface entries for the IA-station example look like?

Example:

IETF Interface-list contains at least twelve entries:

End station component 1:

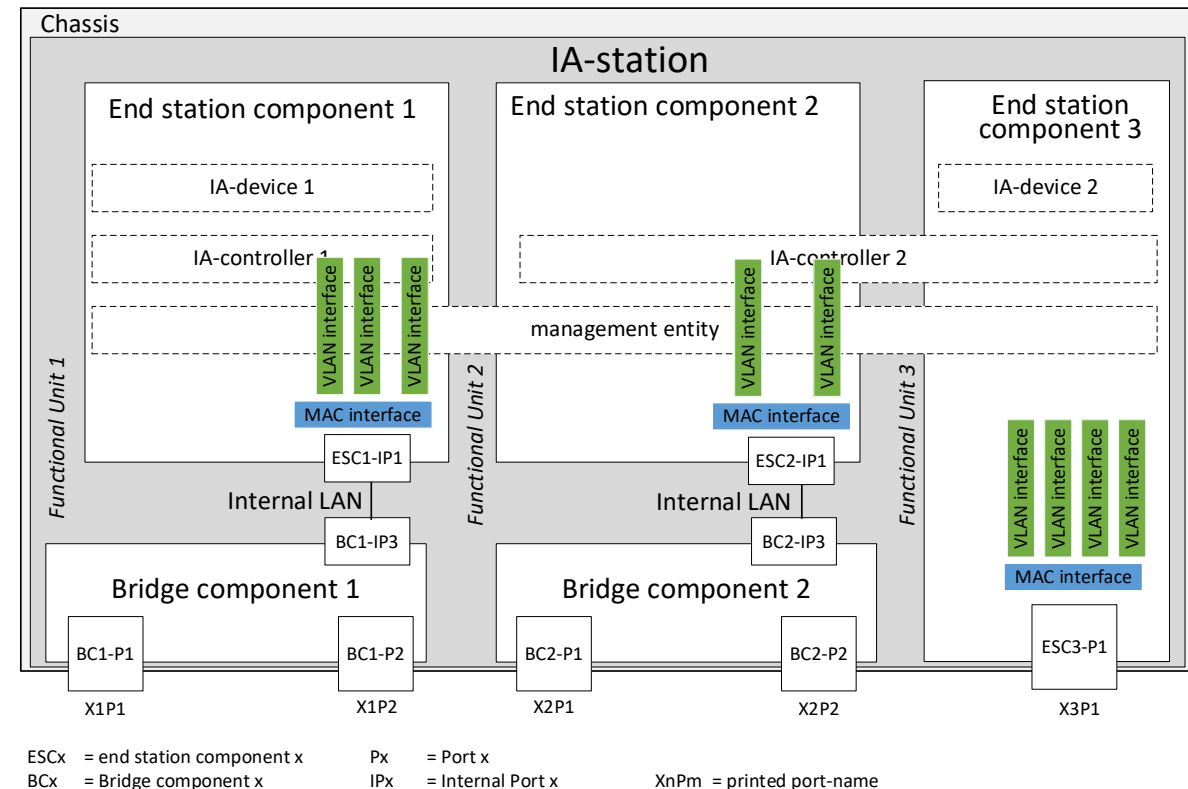
- MAC interface
- Three different VLAN interfaces

End station component 2:

- MAC interface
- Two different VLAN interfaces

End station component 3:

- MAC interface
- Four different VLAN interfaces



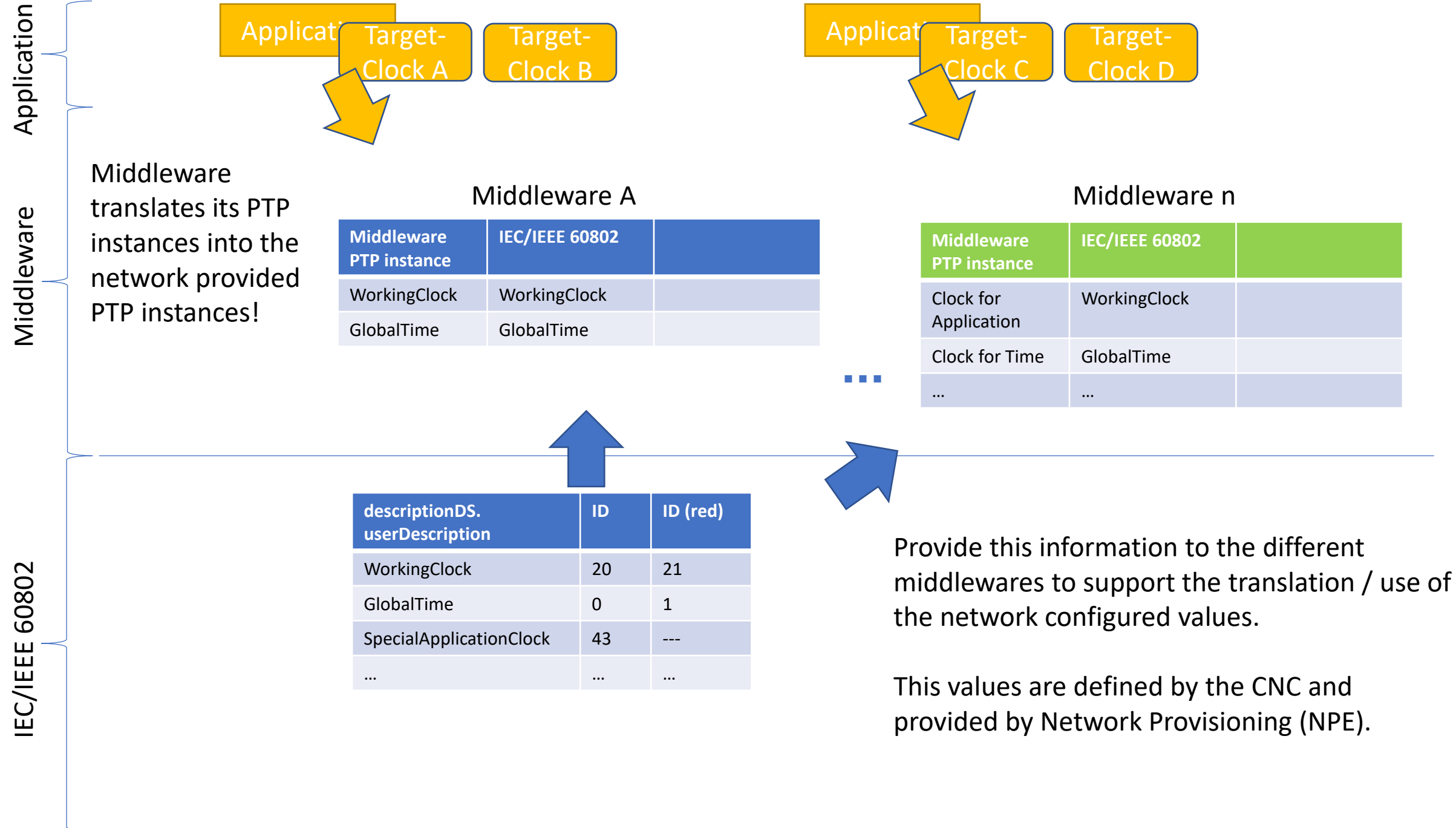
Conclusion - Traffic Type Translation

A translation table which allows late binding of the middleware to the network resources already exists.

Profiling the content of name and description would allow the required translation for TC, VID, PCP and if needed DSCP.

CNC/NPE configured entries IETF Interface-list should solve the problem.

PTP Instance ID translation



IEEE 1588e / .1ASdn YANG descriptionDS.userDescription

Decoupling for the to be used PTP instance could be done following the same principle.

Problem:

IETF interface-list doesn't have an defined ifType for gPTP.

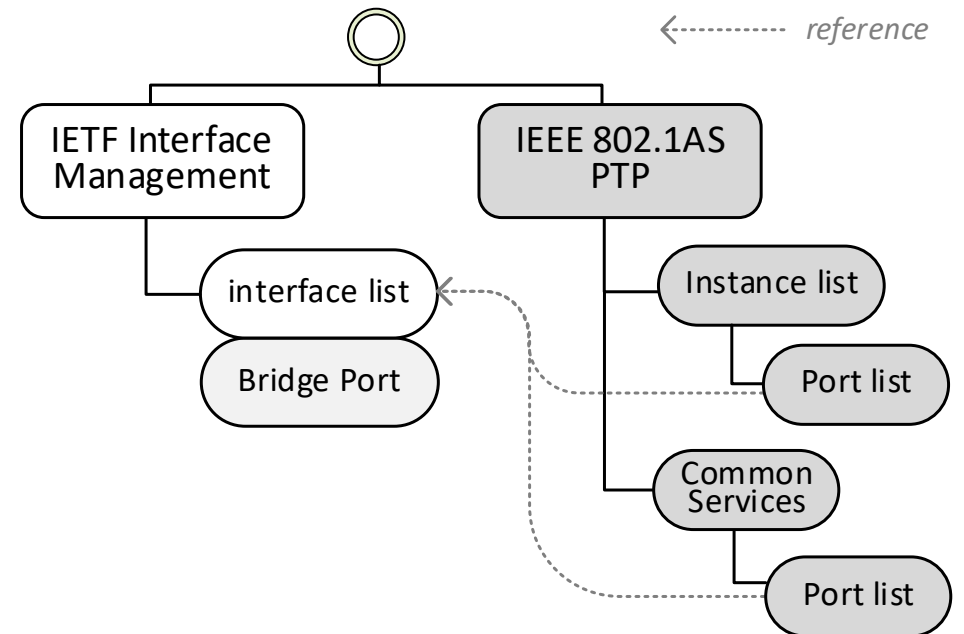
Temporary solution:

Content of name and/or description string (descriptionDS.userDescription) shall be profiled by the TSN-IA profile.

Example:

userDescription :=

- List of clock type e.g. WorkingClock and/or GlobalTime



Sanity check

How could a translation based on the descriptionDS.userDescription entry for the IA-station example look like?

Example:

PTP instance list contains at least four entries:

End station component 1:

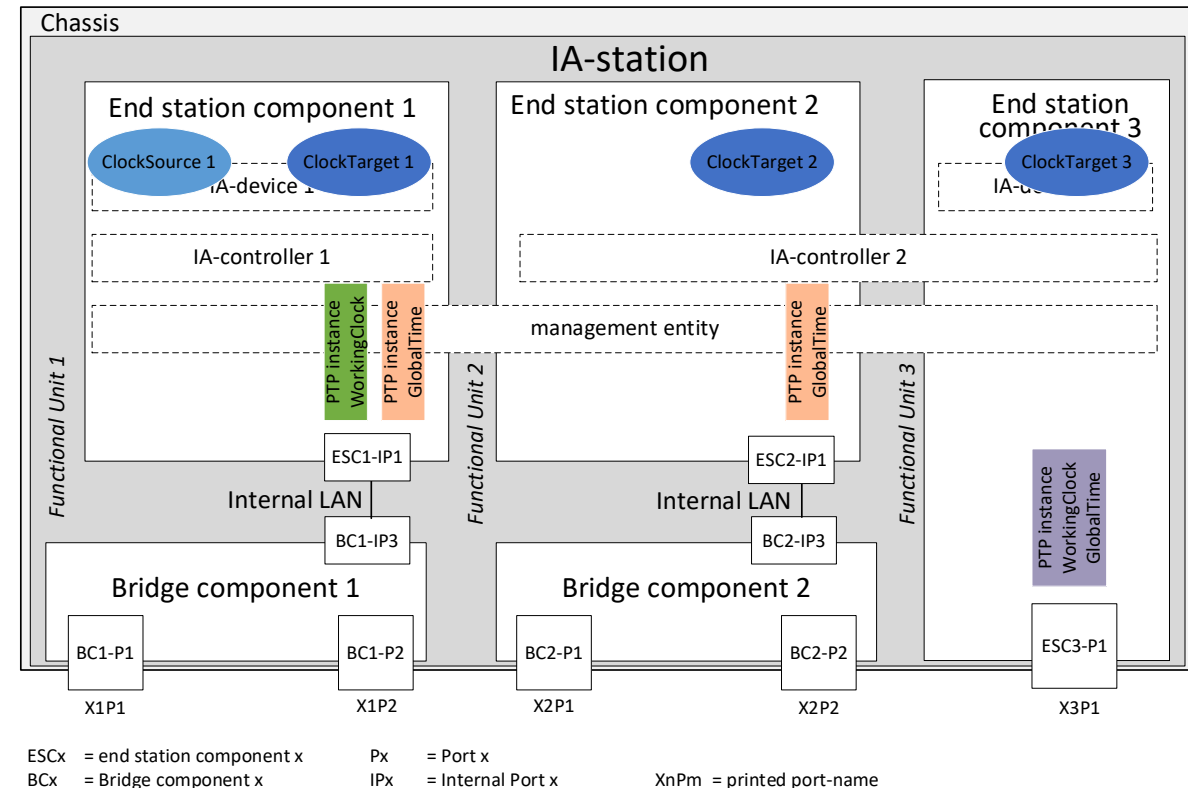
- Two PTP instances, one for WorkingClock and one for GlobalTime

End station component 2:

- One PTP instance for GlobalTime

End station component 3:

- One PTP instances for WorkingClock and GlobalTime



Conclusion - PTP Instance ID translation

Profiling named references instead of instanceID in the 60802 profile gives automation system providers more freedom.

If no ifType for the IETF interface-list will be defined by 1588e/.1ASdn, then the string “descriptionDS.userDescription” could be used for the profiling.

Open:

Binding between ClockSources or ClockTargets and PTP instances seems at the moment not covered by the 1588e/.1ASdn drafts.

Conclusion

Splitting network configuration from middleware configuration could be configured by existing/in definition YANG modules.

Late binding even for PTP instances could be done.

CNE/NPE decides which PTP instance/ PTP instanceID shall be used for what at the IA-stations.

Questions?