BELDEN
SENDING ALL THE RIGHT SIGNALS

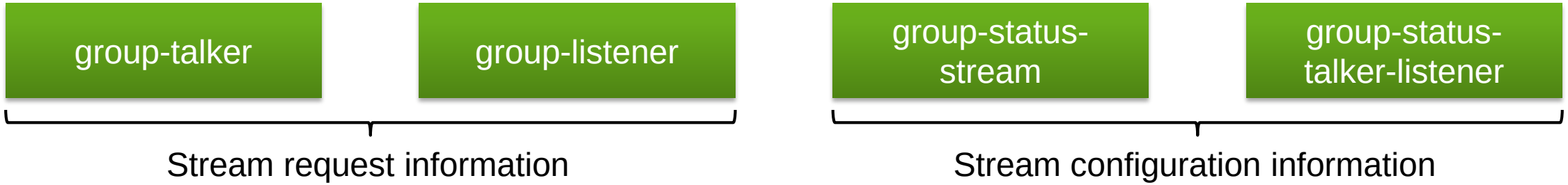
TSN Configuration Enhancements

Stephan Kehrer
Hirschmann Automation and Control GmbH

March 2019

IEEE 802.1Qcc-2018

- A 10,000 foot overview
  - 3 configuration models (Clause 46.1.2): Fully centralized, fully distributed, centralized network/distributed user (*aka hybrid*)
  - Managed objects (Clause 12.32.1-3) and MIBs (esp. Clause 17.7.25) for centralized model
  - MSRPv1 (Clause 35), managed objects (Clause 12.32.4), and MIBs (esp. Clause 17.7.25) for hybrid model
  - TSN UNI data structures (Clause 46.2) & YANG module (Clause 46.3) with 4 core elements:



  - … and some normative text:

"If a YANG-based protocol is specified **by another standard** for the TSN user/network configuration information, that specification shall use the YANG module specified in 46.3.1."

Source: IEEE 802.1Qcc Clause 46.3

IEEE 802.1Qcc and the Fully Centralized Model

Basis for YANG-based protocol:
ieee802-dot1q-tsn-types

Basis for SNMP:
esp. IEEE8021-SR-RM-MIB

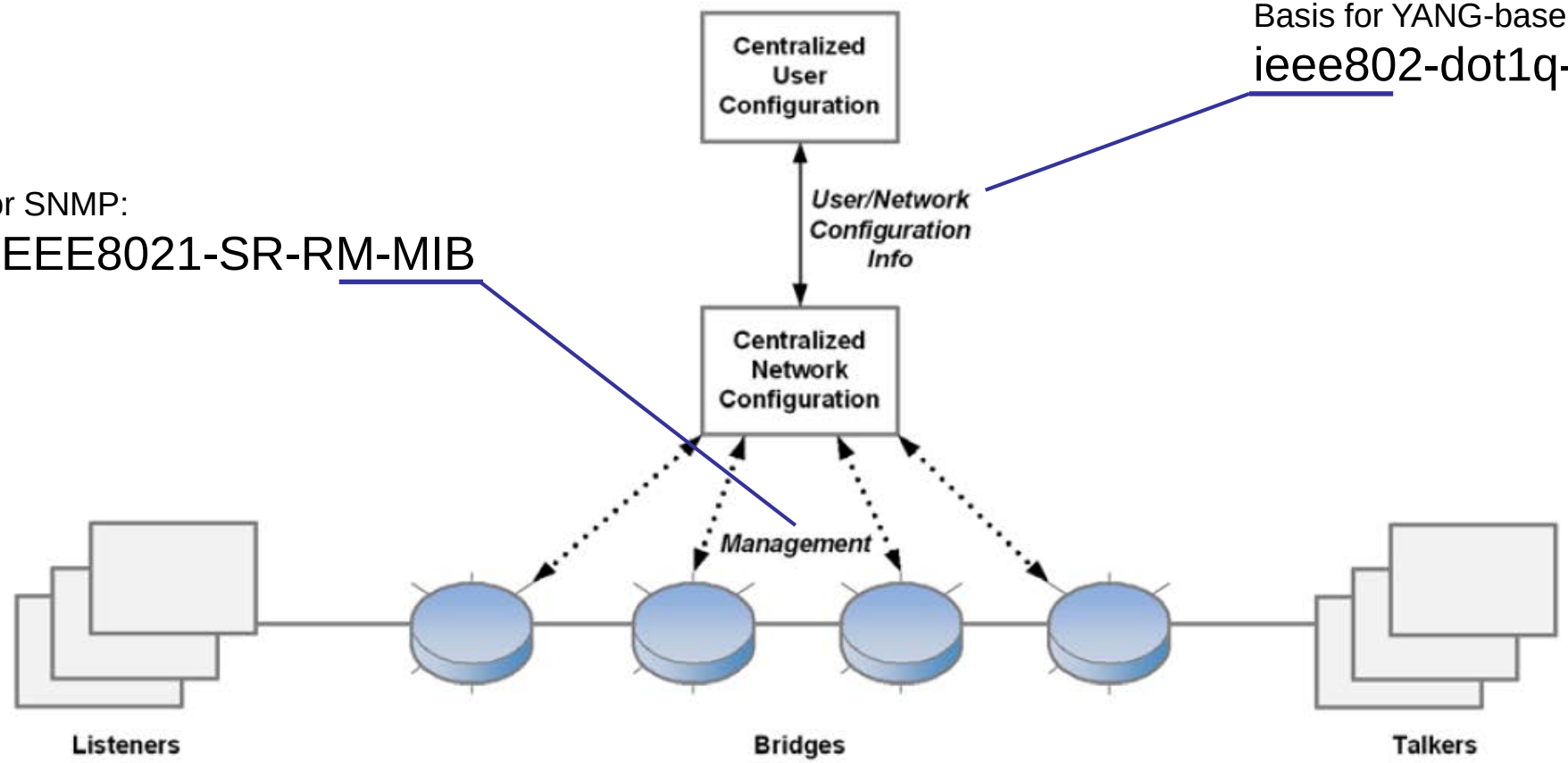


Figure 46-3 — Fully Centralized Model

If we were to use e.g. RESTCONF between CNC & CUC, we would be done…

… not so fast!

Desired state:

draft-tsn-uni.yang

Pyang Validation

```
No warnings or errors
```

Pyang Output

```
module: draft-tsn-uni
    +--rw tsn-uni
       +--rw request-list* [stream-id]
       |  +--rw stream-id          tsn:stream-id-type
       |  +--rw talker
```

Actual state:

ieee802-dot1q-tsn-types.yang

Pyang Validation

```
ieee802-dot1q-tsn-types.yang:1: warning: RFC 6087: 4.1: the module name should start with the string
```

Pyang Output

```
No warnings or errors
```

Source: yangvalidator.com

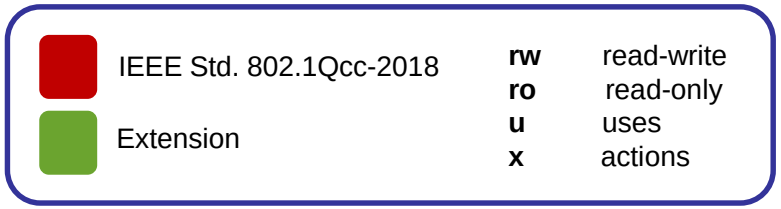
- IEEE 802.1Qcc defines general data structures, but does **not** hook into a root container

Need for an encapsulating YANG module to act as an interface!

Example of an encapsulating YANG module (Simplified)

```
+--rw tsn-service-uni
|     +--rw stream-list* [stream-id]
|        +--rw stream-id
|        +--rw request
|        |  +--rw talker
|        |  |  +--u tsn:group-talker
|        |  +--rw listener-list* [index]
|        |  |  +--rw index
|        |  |  +--u tsn:group-listener
|        |  +---x compute-request
|        +--ro configuration
|           +--u tsn:group-status-stream
|           +--ro talker
|           |  +--u tsn:group-status-talker-listener
|           +--ro listener-list* [index]
|           |  +--ro index
|           |  +--u tsn:group-status-talker-listen
|           +---x deploy-configuration
|           +---x undeploy-configuration
|           +---x delete-configuration
+---x compute-all-requests
+---x deploy-all-configurations
+---x undeploy-all-configurations
+---x delete-all-configurations
```

- Encapsulating YANG module using IEEE 802.1Qcc data structures
- Extended with structure and actions to operate on data
- Consists of two sections:
  - Request = CNC Input
  - Configuration = CNC Output



Fully Centralized Model with requested enhancements

Basis for YANG-based protocol:
ieee802-dot1q-tsn-types
ieee802-dot1q-tsn-service-uni

Basis for SNMP:
esp. IEEE8021-SR-RM-MIB

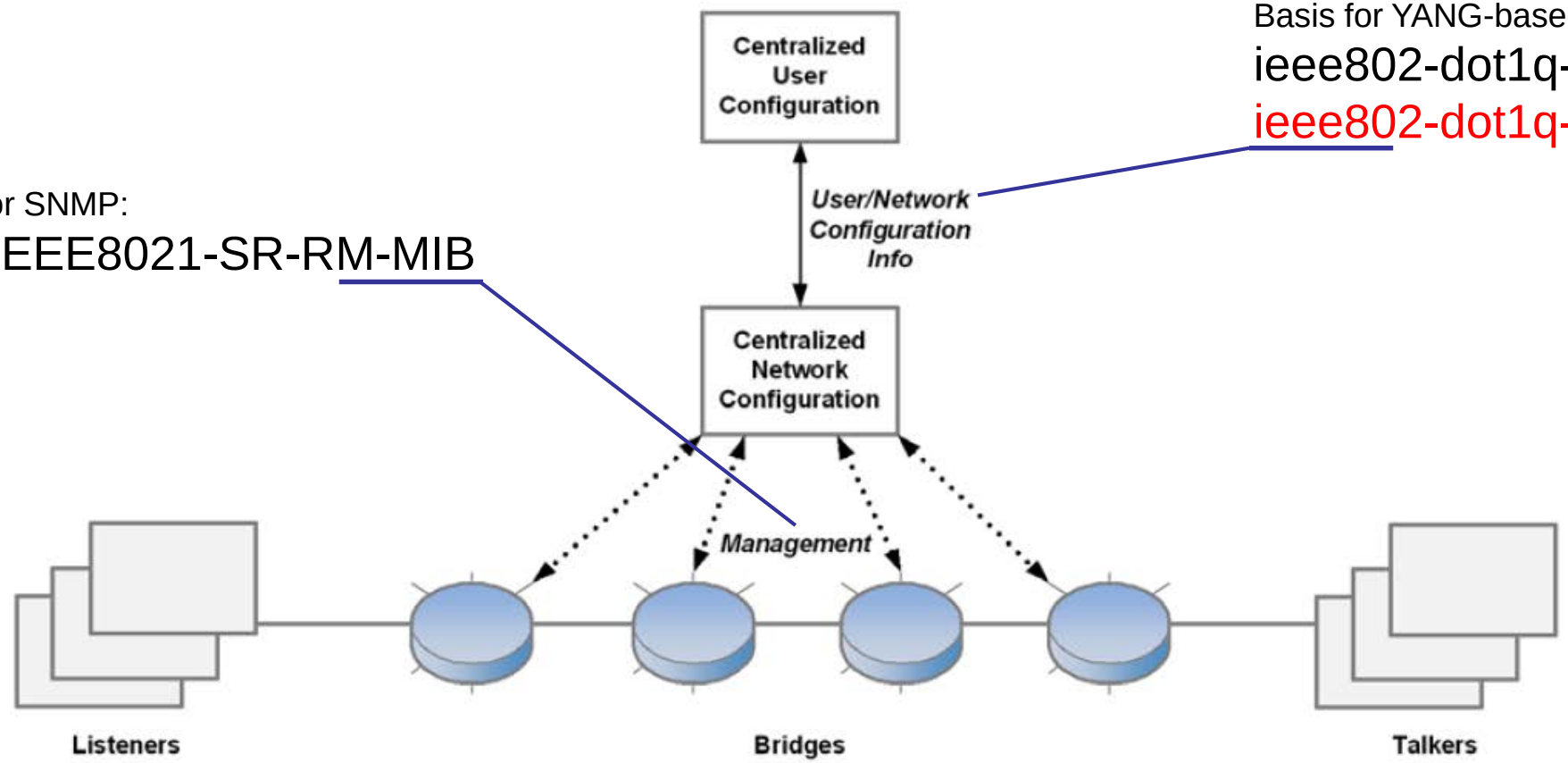
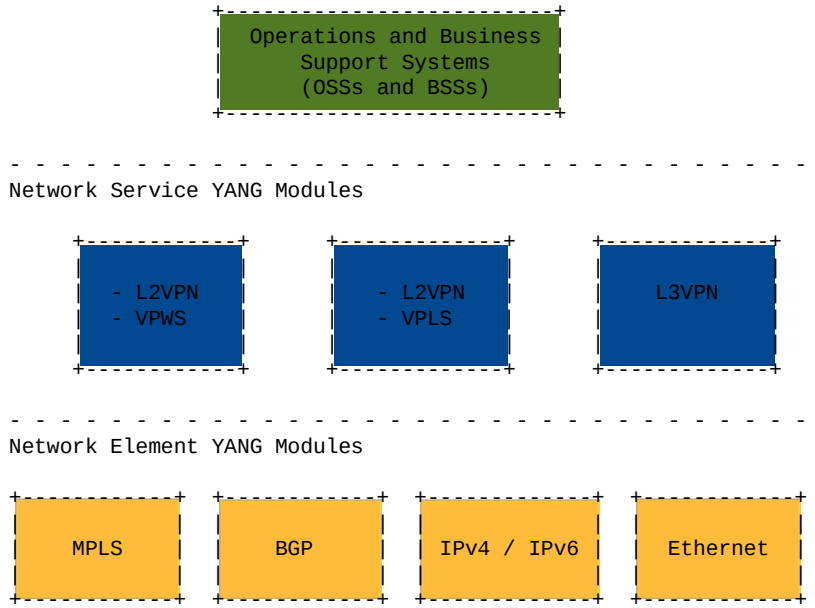


Figure 46-3 — Fully Centralized Model

Why tsn-service-uni?

IETF RFC 8199: **YANG Module Classification**



L2VPN: Layer 2 Virtual Private Network
L3VPN: Layer 3 Virtual Private Network
VPWS: Virtual Private Wire Service
VPLS: Virtual Private LAN Service

Figure 1: YANG Module Abstraction Layers

IEEE 802.1Qcc

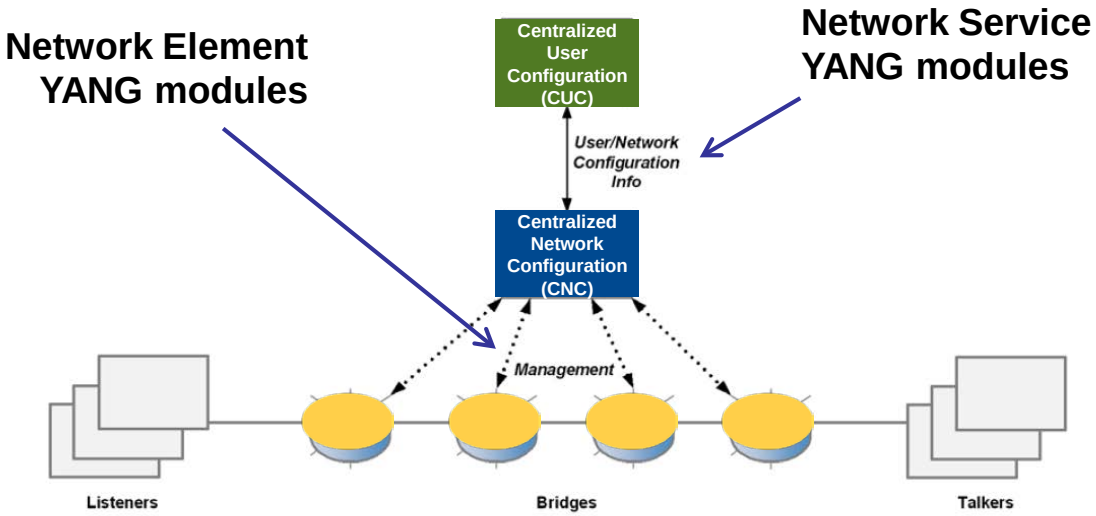


Figure 46-3 — Fully Centralized Model

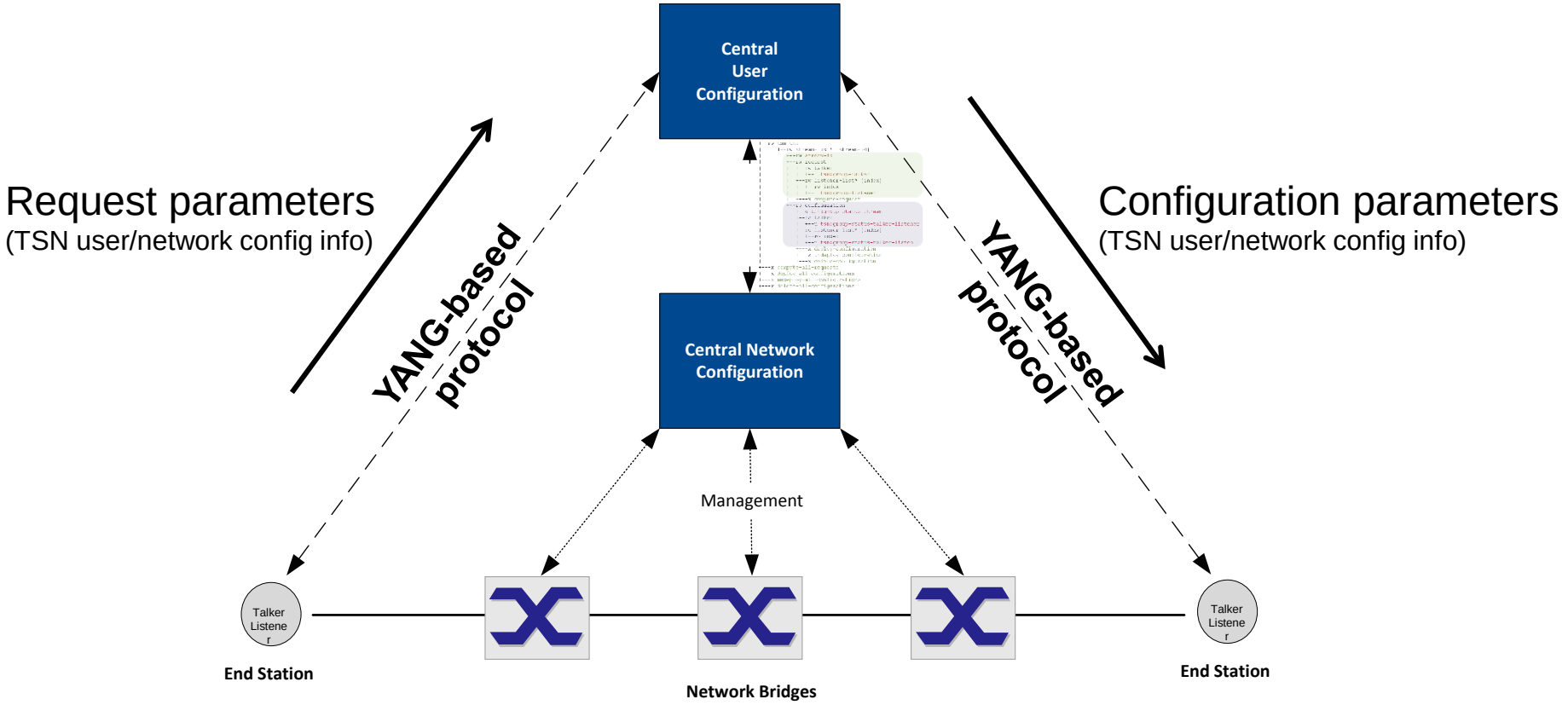
IETF RFC 8199 Section 2:

„**Network Element** YANG Modules describe the configuration, state data, operations, and notifications of **specific device-centric technologies or features**."

„**Network Service** YANG Modules describe the configuration, state data, operations, and notifications of **abstract representations of services** implemented on *one or multiple network elements*."

Is there also a different, second TSN UNI?

For the sake of this discussion, let's assume a YANG-based protocol was used for end-stations as well…



"If a YANG-based protocol is specified **by another standard** for the TSN user/network configuration information, that specification shall use the YANG module specified in 46.3.1." *Source: IEEE 802.1Qcc Clause 46.3*

TSN End-Station UNI and Relationship to IEEE 802.1Qcc/TSN Service UNI

tsn-service-uni: **(stream-centric)**

```
+--rw tsn-service-uni
|     +--rw stream-list* [stream-id]
|        +--rw stream-id
|        +--rw request
|        |  +--rw talker
|        |  |  +--u tsn:group-talker
|        |  +--rw listener-list* [index]
|        |  |  +--rw index
|        |  |  +--u tsn:group-listener
|        |  +---x compute-request
|        +--ro configuration
|           +--u tsn:group-status-stream
|           +--ro talker
|           |  +--u tsn:group-status-talker-listener
|           +--ro listener-list* [index]
|           |  +--ro index
|           |  +--u tsn:group-status-talker-listen
|           +---x deploy-configuration
|           +---x undeploy-configuration
|           +---x delete-configuration
+---x compute-all-requests
+---x deploy-all-configurations
+---x undeploy-all-configurations
+---x delete-all-configurations
```

IEEE Std. 802.1Qcc-2018

CNC Input

CNC Output

rw	read-write
ro	read-only
u	uses
x	actions

tsn-end-station-uni:
(device role-centric)

Talker

```
Stream
  stream-id
  stream-request
  +--u tsn:group-talker
  stream-configuration
  +--u tsn:group-status-talker-listener
```

Listener

```
Stream
  stream-id
  stream-request
  +--u tsn:group-listener
  stream-configuration
  +--u tsn:group-status-talker-listener
```

ES

Talker

CUC/CNC

Talker

```
Stream
  stream-id
  stream-request
  +--u tsn:group-talker
  stream-configuration
  +--u tsn:group-status-talker-listener
```

Listener

```
Stream
  stream-id
  stream-request
  +--u tsn:group-listener
  stream-configuration
  +--u tsn:group-status-talker-listener
```

ES

Listener

IEEE 802.1Qcc and the Fully Centralized Model

Basis for YANG-based protocol:
ieee802-dot1q-tsn-types
some-org-tsn-end-station-uni

Basis for SNMP:
esp. IEEE8021-SR-RM-MIB

Basis for YANG-based protocol:
ieee802-dot1q-tsn-types
ieee802-dot1q-tsn-service-uni

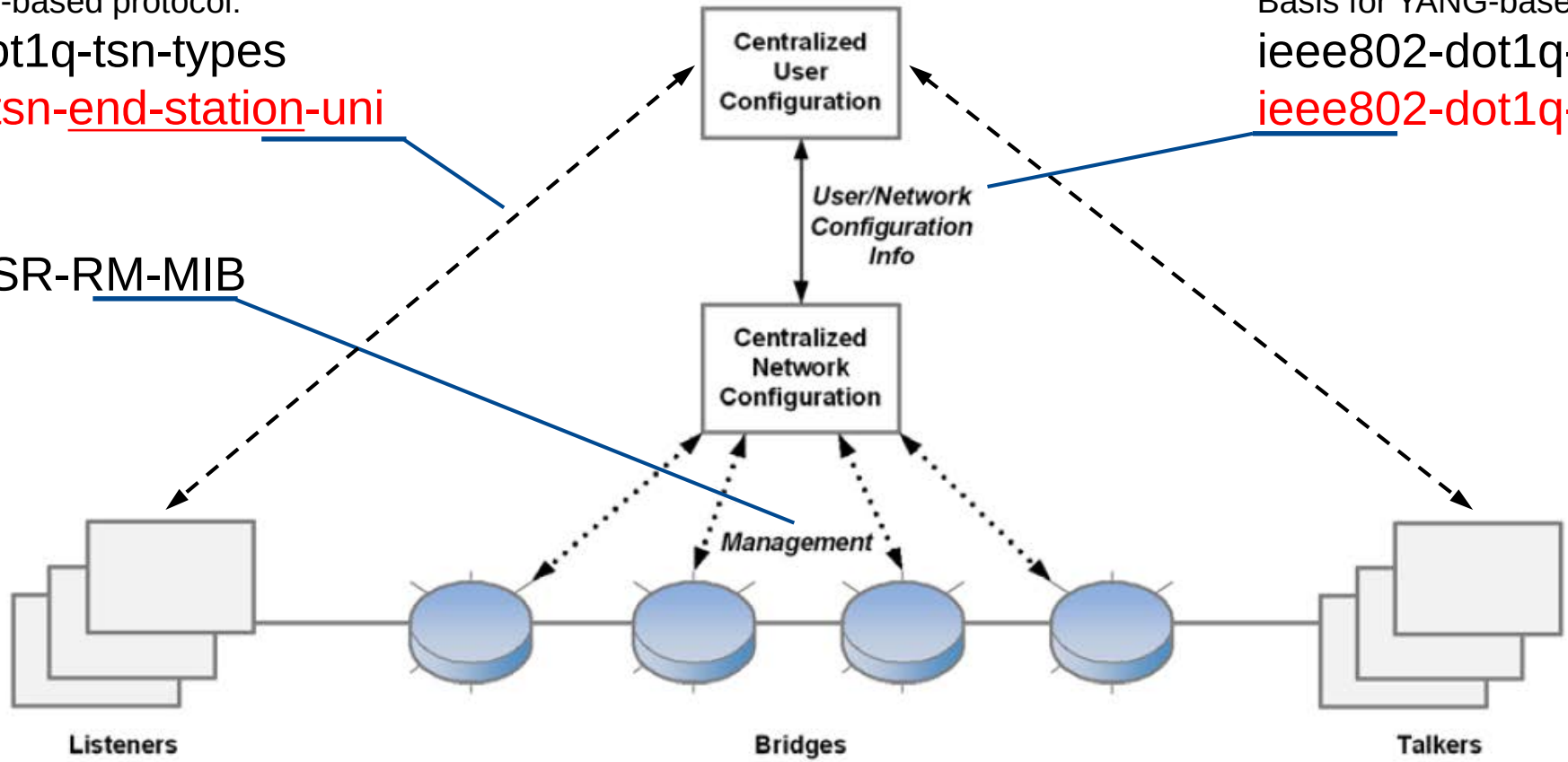


Figure 46-3 — Fully Centralized Model

Yes, there are two different UNIs, but they should use the same underlying data structures.
Why? → This also leads towards harmonizing the different configuration models!

IEEE 802.1Qcc and the Hybrid Model

Basis for YANG-based protocol:
ieee802-dot1q-tsn-types
ieee802-dot1q-tsn-end-station-uni

Basis for YANG-based protocol:
ieee802-dot1q-tsn-types
ieee802-dot1q-tsn-service-uni

Basis for SNMP:
esp. IEEE8021-SR-RM-MIB

Protocol used by Endstation:
LRP/RAP

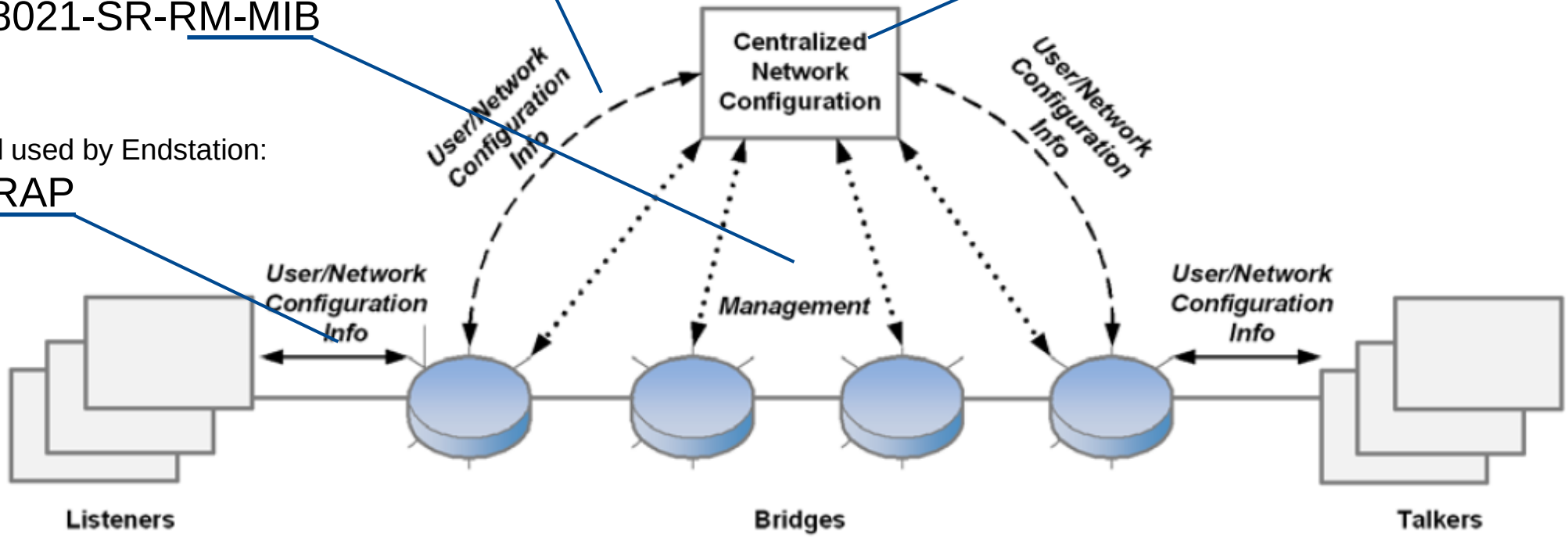


Figure 46-2 — Centralized Network / Distributed User Model

IEEE 802.1Qcc and the Fully Distributed Model

Protocol used for network setup:
LRP/RAP

Protocol used by Endstation:
LRP/RAP

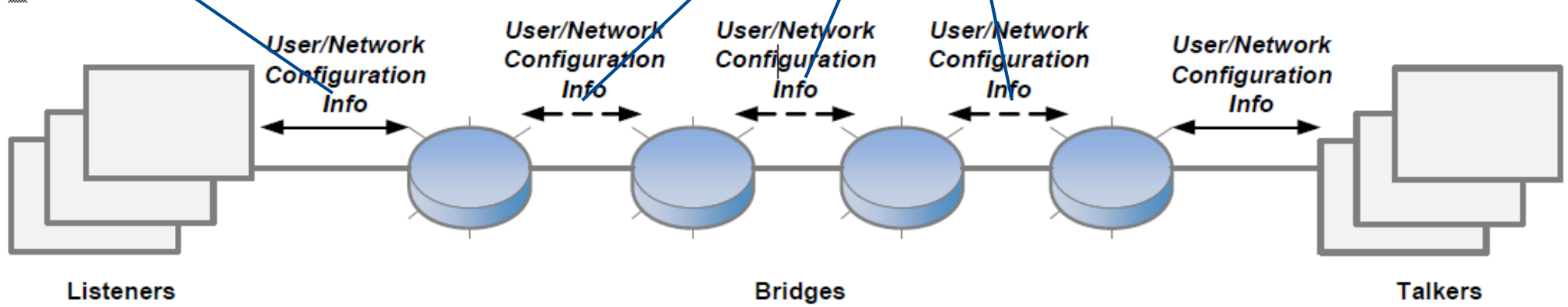


Figure 46-1—Fully distributed model

What is the proposal of this presentation?

Proposal

- Start to work on a PAR and CSD for TSN configuration enhancements in order to:
  - define a tsn-service-uni that can be used as interface between CNC and CUC for both, the centralized and hybrid model
  - define a tsn-end-station-uni to pass information from a bridge to a CNC in the hybrid model
    - this is intended to enable configuration of a network by endstations using RAP/LRP without the knowledge of how the network is managed (decentralized or hybrid)
  - allow a complete configuration workflow for configuring TSN networks
    - **This means** enhancing the existing UNI definition with missing pieces required for a complete configuration
    - **This does not mean** enhancing the existing UNI definitions with functionality that is specific to a certain application or use case – such enhancements should be done in application specific YANG modules outside of IEEE

BELDEN
SENDING ALL THE RIGHT SIGNALS

Thank you