Frank Bär, Dr. Frank Breiter, Alexander Groß

Enhancing TSN Configuration: A Hybrid CORECONF-NETCONF Approach for Constrained Devices

Fraunhofer IPMS

Fraunhofer-Institut für Photonische Mikrosysteme IPMS

Gefördert durch:

Bundesministerium für Forschung, Technologie und Raumfahrt

Motivation

Why do we need new approaches to device configuration?

Challenge

- Ensuring reliable, scalable deployment methods across thousands of diverse, time-sensitive devices
- TSN (Time-Sensitive Networking) is essential, but configuration must keep pace with complexity

Industry 4.0 & Ethernetification

- Rapid increase in smart, connected devices
- **Ethernet** as the unified backbone for industrial connectivity
- Exponential growth drives configuration complexity

IT/OT Convergence

- **IT** systems integrating deeply with Operational Technology (**OT**)
- Examples: CNC machines, industrial robots, PLCs, sensors, actuators
- OT requires strict Quality of Service (**QoS**) and high reliability
- Heterogeneous networks: Diverse device capabilities, protocols, and requirements

Smart Factory Evolution

- Real-time data exchange between machines, systems, and sensors
- Automated workflows and adaptive manufacturing processes
- Interconnected devices require precise time synchronization and configuration
- Need for scalability and flexibility in device management to support continuous innovation

Configuration protocols

Short Overview

Common approaches to network configuration

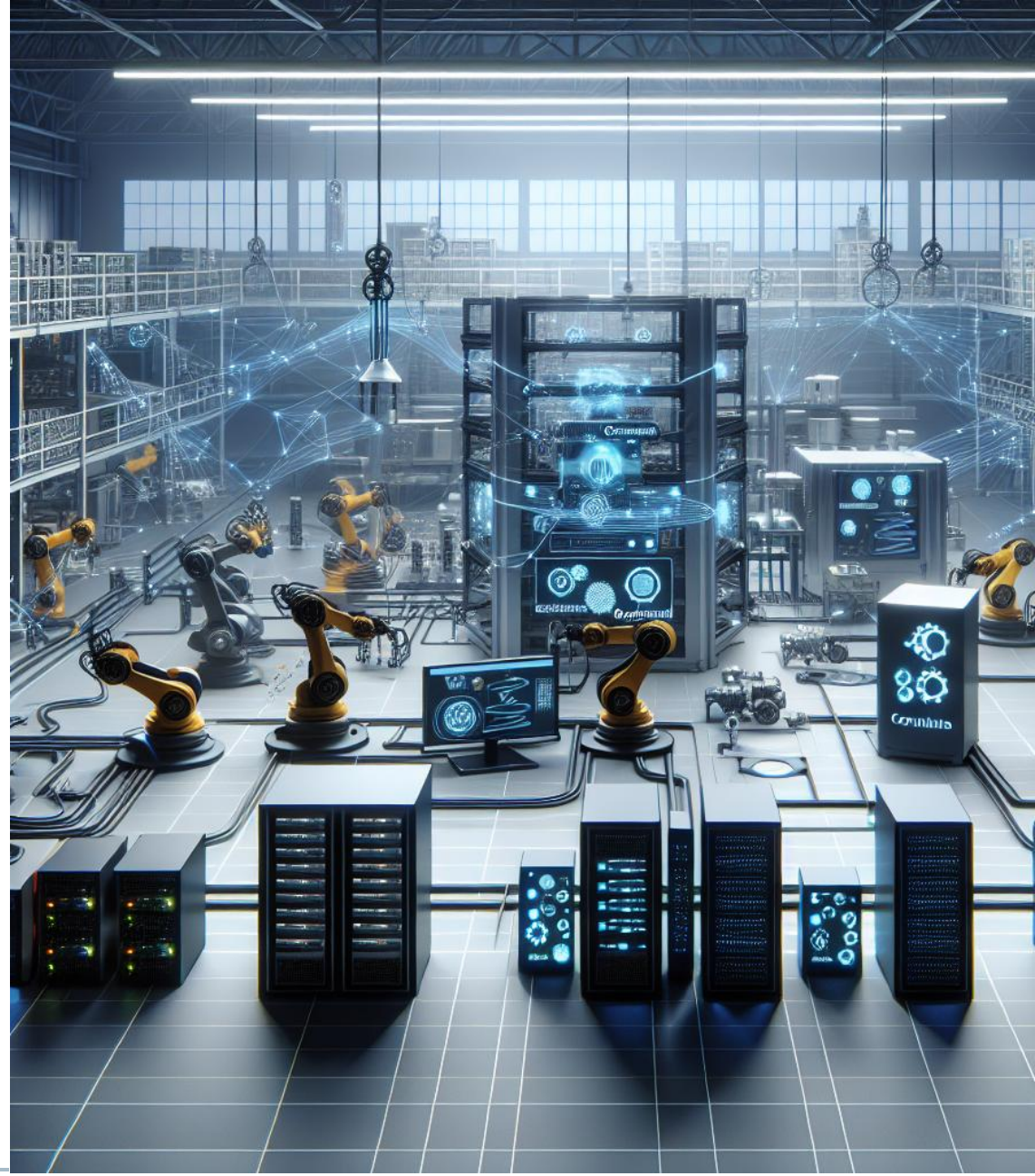
- current devices often use proprietary configuration protocols (CLI, REST)
- Simple Network Management Protocol (SNMP) was originally intended for network configuration purposes, but is only used for monitoring and status data
- SNMP lacks complexity and integrity
- Other missing features: atomic operations, transactional configuration, error handling

The Future speaks NETCONF/YANG

- A standard with strong traction in “feature-rich” devices, but not fully saturated in the market
  - YANG supports complex network modelling
  - NETCONF provides extended Protocol Features (e.g. candidates, error-options)

However…

- NETCONF (protocol operation, XML-parsing, …) is a heavy task demanding lots of ressources
- constrained devices are still often statically configured/need machine access for reconfiguration



Configuration protocols

CORECONF – CoAP Management InterfaceS

What is CORECONF?

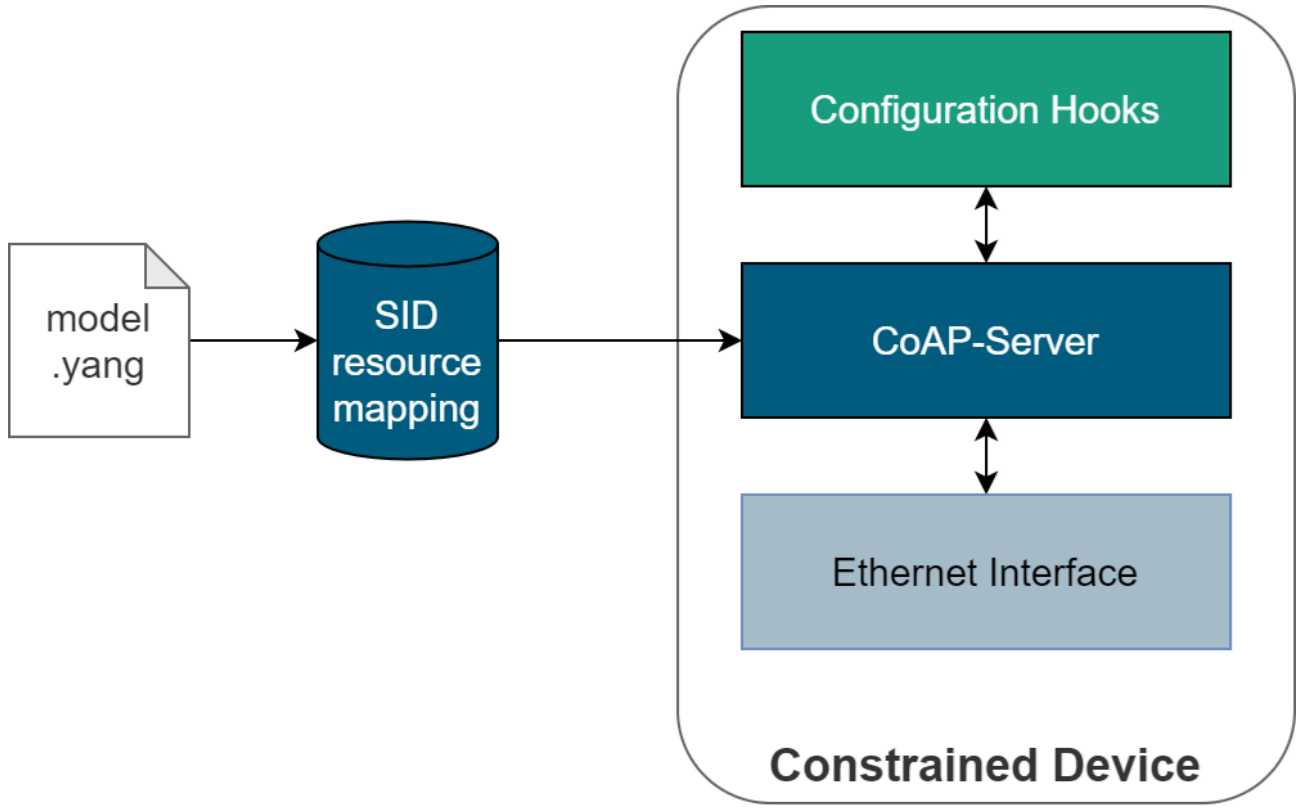
- REST-like interface via CoAP-Server
- YANG-based data model (SID - Schema Item iDentifiers)

General advantages

- CoAP is a well-defined RFC-Standard tailored to constrained environments
- CORECONF makes use of existing YANG models, same as NETCONF

Advantages for constrained devices

- Binary data representation (CBOR)
- Simple semantics for data access (GET, PUT, ...)
- MAY include datastores, but not required



IEC/IEEE 60802

TSN Profile for Industrial Automation

IEEE60802 "defines time-sensitive networking profiles for industrial automation. The profiles select **features, options, configurations, defaults, protocols, and procedures** of bridges, end stations, and LANs to build industrial automation networks."*

Device classification

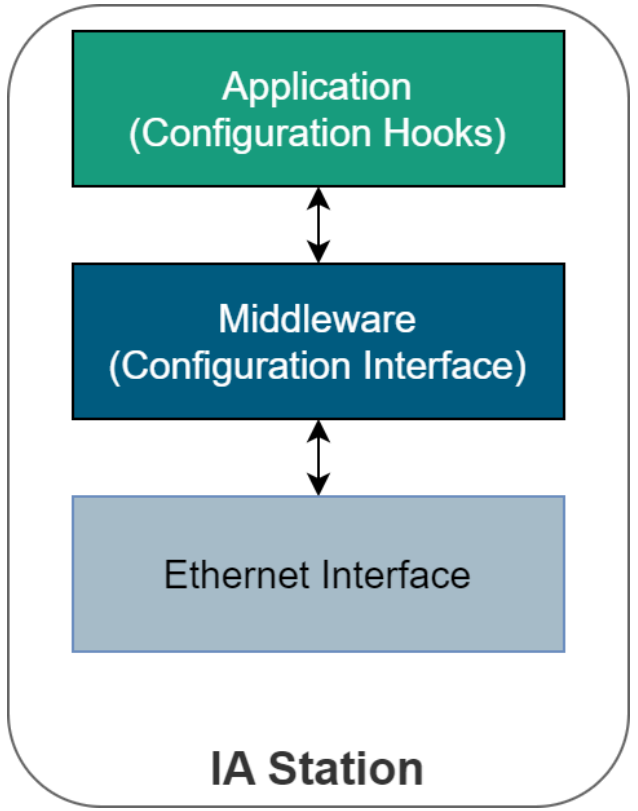
- Everything is an **IA-station**
- Conformance classes:
  - Feature rich **ccA**
    - e.g. PC, PLC
  - Constrained **ccB**
    - e.g. sensors, actors

IEEE60802 configuration

- Restricted to NETCONF/YANG or RESTCONF/YANG for ccA-devices

Recent draft 3.4

- Configuration of ccB IA-Stations (device interface, protocol) is **not specified**
- Introduction of management proxies



* Source: IEC/IEEE 60802 TSN Profile for Industrial Automation |

IEEE60802 Management Proxies

Working principle

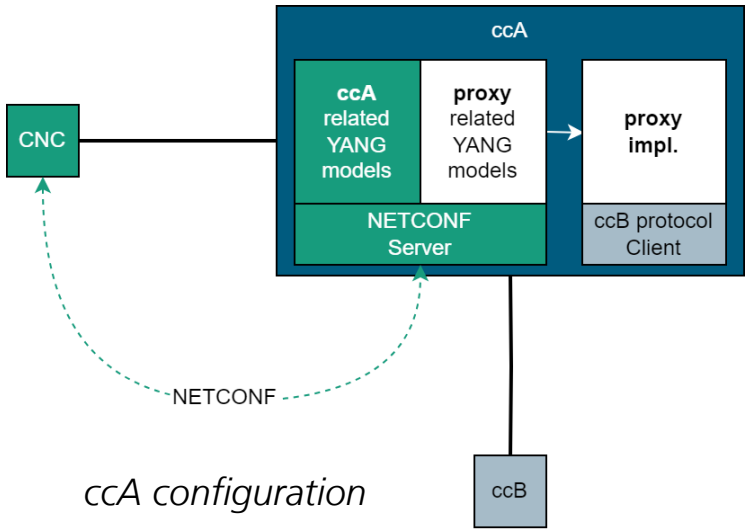
- Extension of IA-Station (ccA) bridges protocol gap for configuration of ccB Devices (acts as a translator)
- once discovered, constrained devices (ccBs) are assigned a proxy by the CNC (based on the proxies capabilities)
- The proxy accepts configuration requests for proxied clients via dedicated YANG models
- Proxy the requests to the desired protocol

Requirement

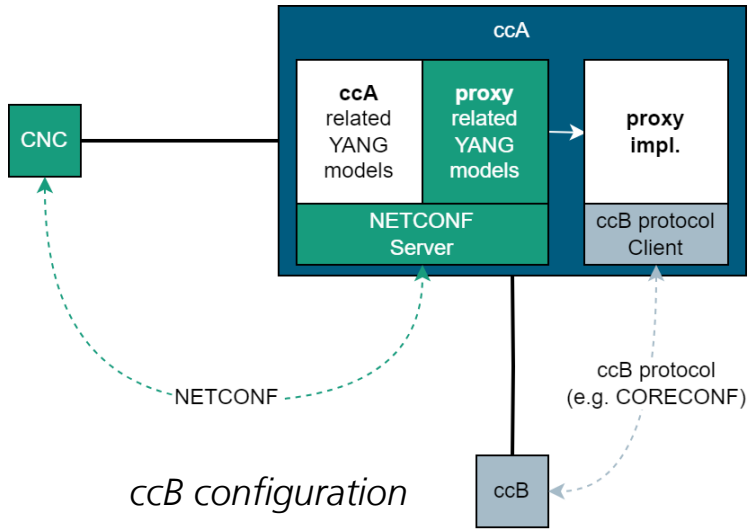
- ccBs protocol must be IP-based

Benefits

- CNC only needs to support NETCONF
- Resource-intensive protocol translation is performed on ccA device



ccA configuration



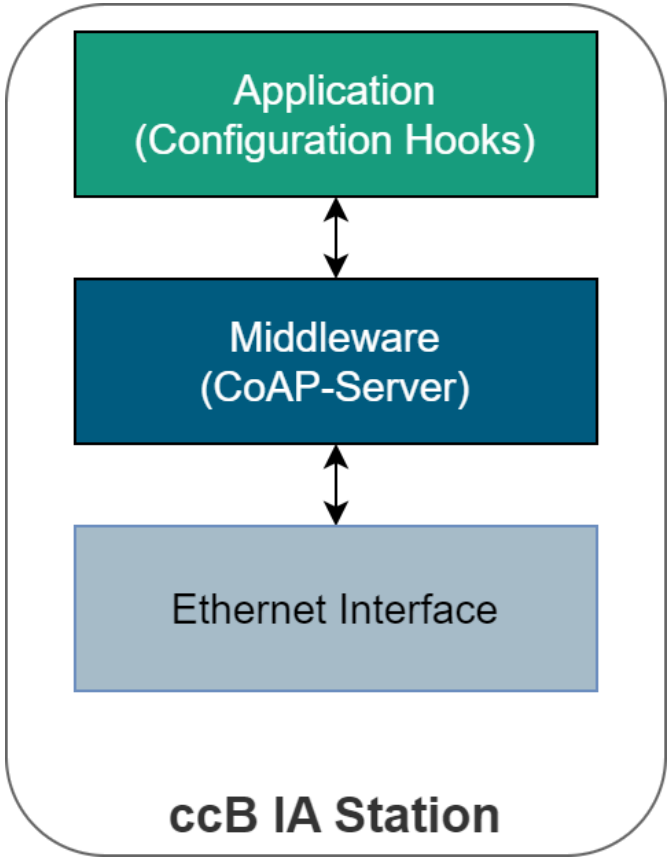
ccB configuration

IEEE60802 & CORECONF

Configuration protocol for constrained devices

Integration of CORECONF with NETCONF

- CORECONF seamlessly integrates with the IEEE60802 architecture
- Utilizing the Management Proxy functionality of IEEE60802
- YANG-based data modelling positions CORECONF as a strong candidate for the standard configuration protocol for ccB devices
- Still allows “standalone” management of a constrained („ccB“) device via CORECONF-CNC



Implementation of a CORECONF-Proxy

Challenge NETCONF / CORECONF Translation

- Ressource-heavy XML-parsing is done on the proxy side
- Constrained devices receive binary coded messages

But one NETCONF request can *simultaneously*:

- Access different YANG models and access different (hierarchical) layers in the same YANG model
- Combine different access modes/operations (replace, merge, delete, ...)
  - → Most NETCONF-requests cannot be mapped to a single CORECONF request

For Example:

- Semantic of some "operations" requires additional read operations in CORECONF
  - e.g. operation="delete" needs to raise an error if no configuration data exists for the resource
  - But a DELETE request in CORECONF always returns (2.02 Deleted) for a valid resources, even if it is empty

Gefördert durch:
Bundesministerium für Forschung, Technologie und Raumfahrt

Implementation of a CORECONF-Proxy

Challenge NETCONF request mapping

- “error-option” behaviour for <edit-config>-operation
  - CORECONF has no concept of error-options
  - Error-options need to be implemented on proxy side
- <get-config>
  - Shall return only configuration data
  - GET-Response needs to be filtered for relevant data

However, there is a gap:

- Management proxies are not only forwarding requests
- proxies needs to have a understanding of what‘s happening on the proxied device:
  - Process requests and responses
  - Check for changed configuration data
  - Handle error-options

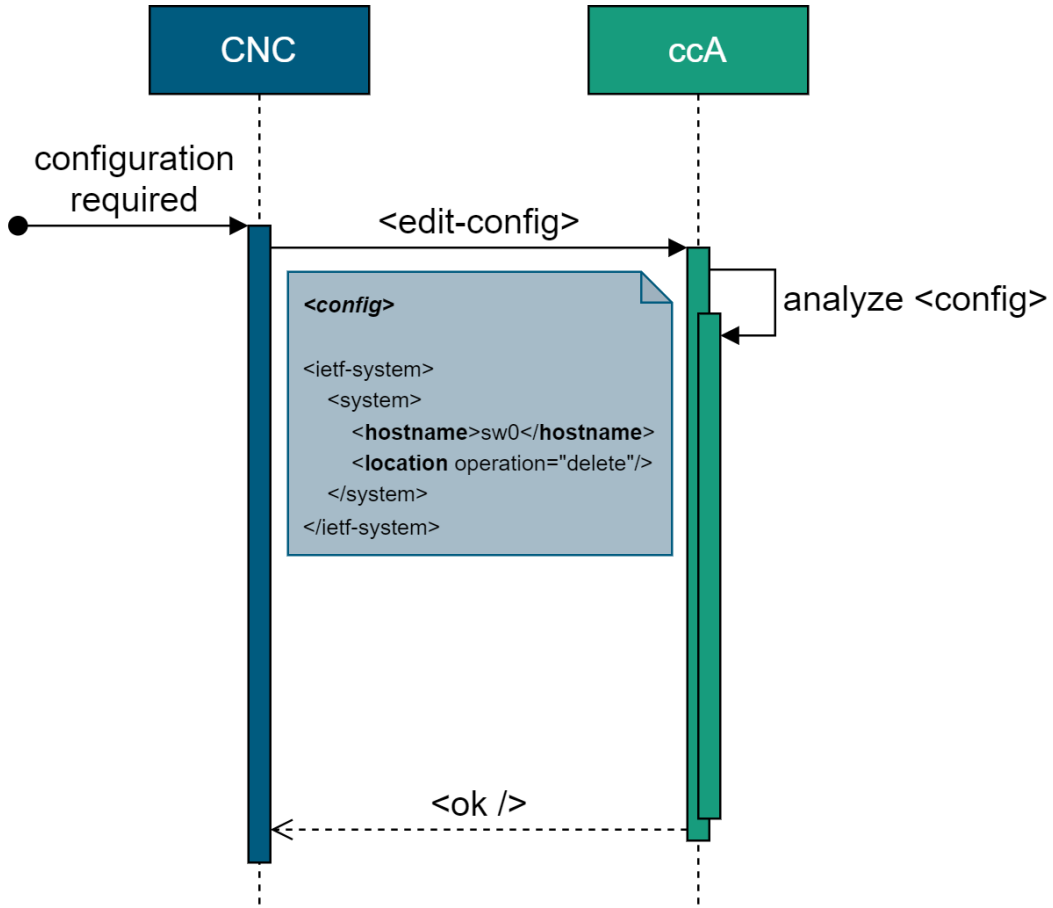
→ The proxy needs to provide datastore access for each individual proxy-target

- **Accessing the specific data and models for a constrained device in the proxies datastore is currently undefined**

Gefördert durch:
Bundesministerium für Forschung, Technologie und Raumfahrt

Implementation of a CORECONF-Proxy

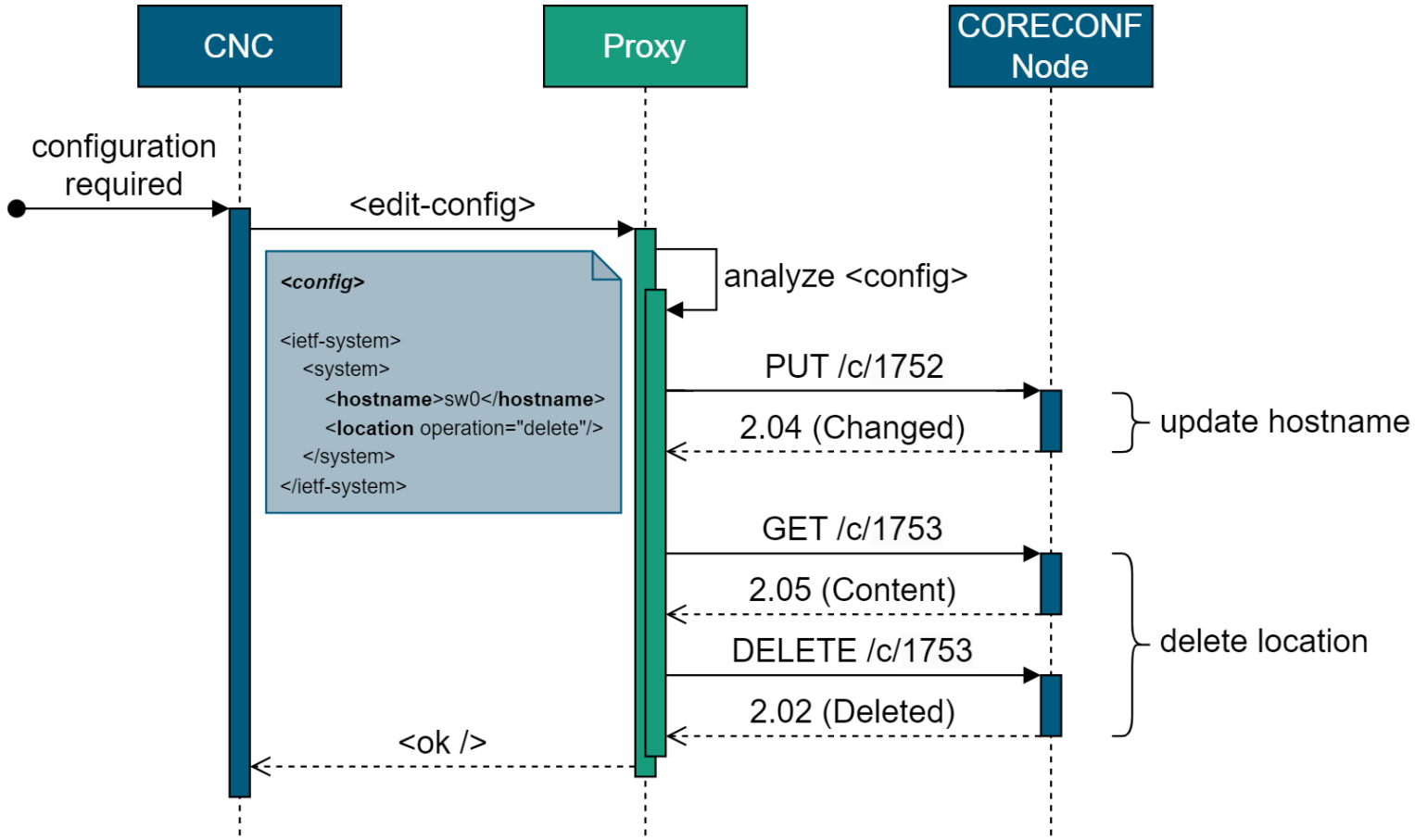
Challenge NETCONF / CORECONF Translation

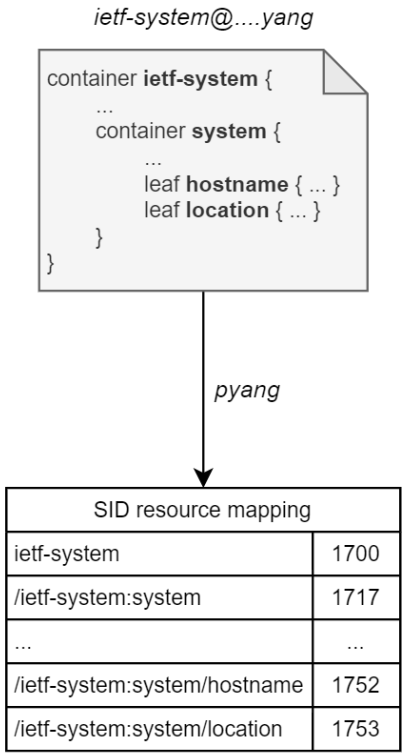


Gefördert durch:
Bundesministerium für Forschung, Technologie und Raumfahrt

Implementation of a CORECONF-Proxy

Challenge NETCONF / CORECONF Translation





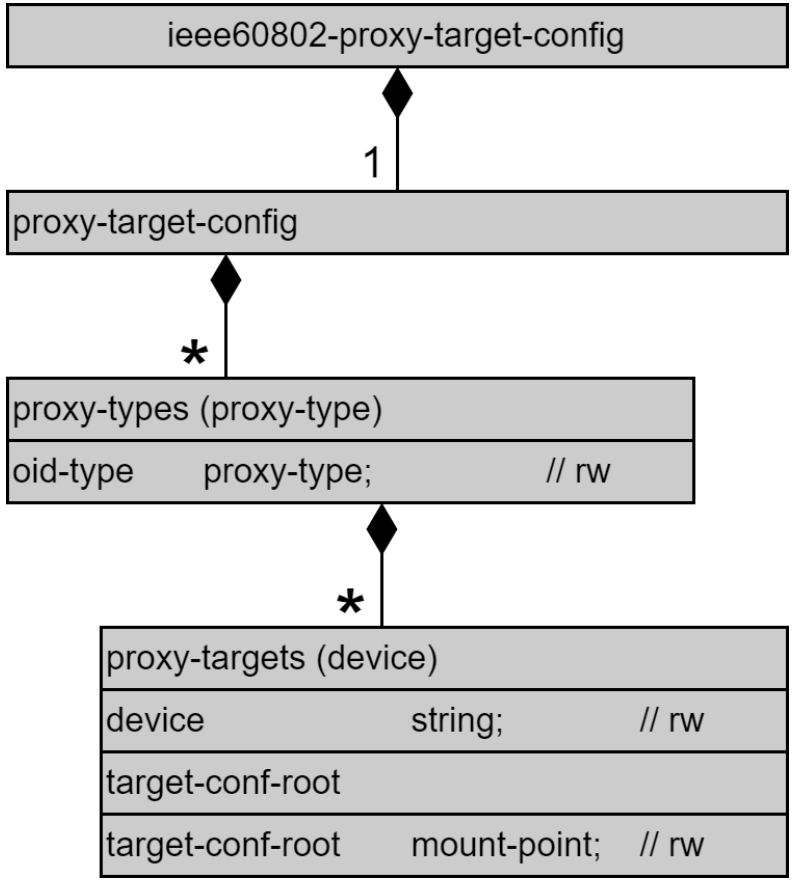
Gefördert durch:
Bundesministerium für Forschung, Technologie und Raumfahrt

IEEE60802 Management Proxies

Proposal: iecieee60802-proxy-target-config

- Same nested list structure as in iecieee60802-proxy
- Configuration data for different proxy-targets stored in separate paths

```
module: iecieee60802-proxy-target-config
  +--rw proxy-target-config!
     +--rw proxy-types* [proxy-type]
        +--rw proxy-type          string
        +--rw proxy-targets* [device]
           +--rw device           string
           +--target-conf-root
              +--mp target-conf-root
```



Gefördert durch:
Bundesministerium für Forschung, Technologie und Raumfahrt

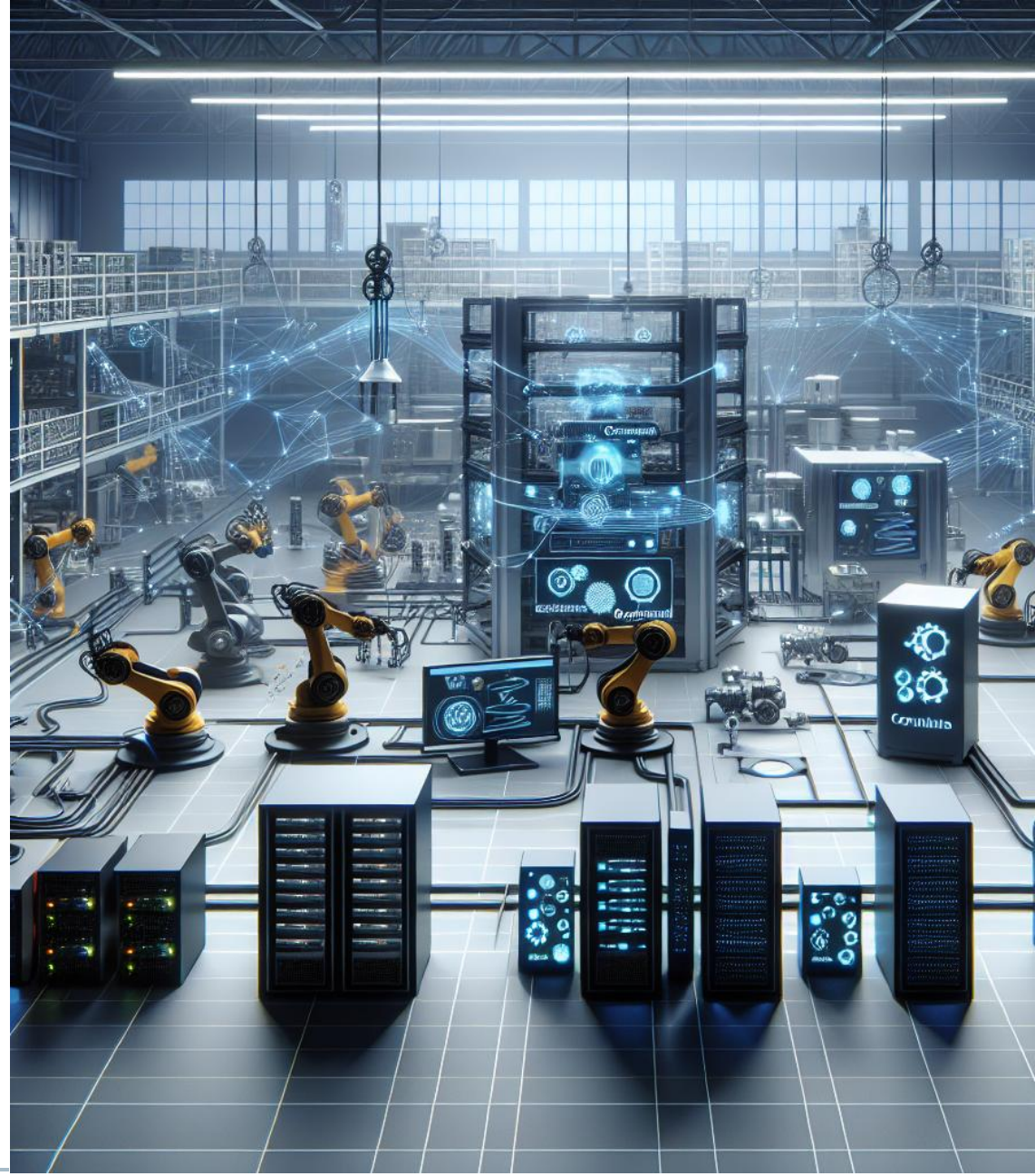
Conclusion + Outlook

Conclusion / Outlook

- **Configuration of constrained devices** remains a problem that requires a general solution
- **CORECONF** is a future-oriented protocol for ccB management
  - Based on the same data models as NETCONF
  - Simpler and less powerful access semantics, but can be well integrated with NETCONF
  - Even CORECONF has limitations

    → There will be devices that are not even capable of running a CORECONF server

- **IEEE60802 management proxies** are a good approach to configuration auf constrained devices
  - We propose an additional YANG model for proxied configuration to fill the knowledge gap on the proxy side



Fraunhofer IPMS

Fraunhofer Institute for Photonic Microsystems IPMS

Contact

Frank Bär
Division DCC
Tel. +49 351 8823-1048
frank.baer@ipms.fraunhofer.de
www.ipms.fraunhofer.de

Benchmarks

Ressource allocation

- Preliminary! Ressource utilization heavily depends on the installed/implemented YANG-models

- Embedded linux NETCONF (ROM / RAM Usage)
  - Libyang, sysrepo, libnetconf2, netopeer2 server: ~2200kB ROM
    - The true ressource utilization comes from linux dependencies like libssh, openssl, curl, IP-stack etc.
  - NETCONF Server (netopeer2-server): ~175MB RAM

- CoAP-Server + CBOR on RTOS:
  - 1950kB ROM / 2,5kB RAM Usage
  - Model implementation adds to ROM/RAM