

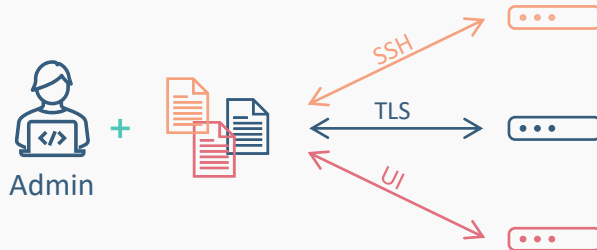
# Evaluation eines modellbasierten Managements in verschiedenen Systemumgebungen

---

Forschungskolloquium 29.07.2021

### Ausgangssituation

Konfiguration einer heterogenen Infrastruktur  
Bsp.: Datacenter

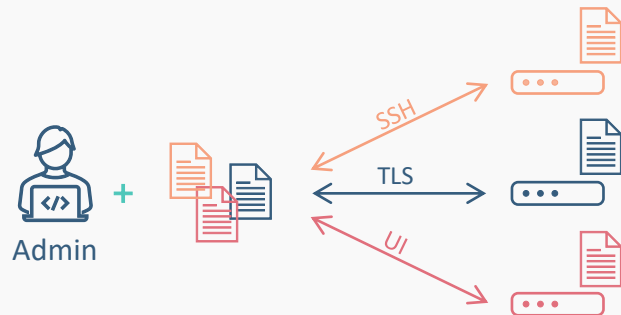


### Probleme

- Gerätespezifische Anpassungen notwendig
- Unterschiedliche Schnittstellen
- Hohes Fehlerpotential
- Kenntnis für jedes Gerät erforderlich

### Ausgangssituation

Konfiguration einer heterogenen Infrastruktur  
Bsp.: Datacenter



### Probleme

- Gerätespezifische Anpassungen notwendig
- Unterschiedliche Schnittstellen
- Hohes Fehlerpotential
- Kenntnis für jedes Gerät erforderlich

### Lösung Part 1: YANG

→ Einheitliches Datenmodell

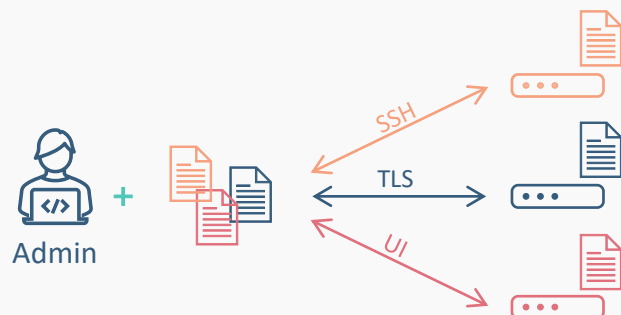


Use Case	Modell
Interfaces	ietf-interfaces
DHCP	ietf-dhcp



### Ausgangssituation

Konfiguration einer heterogenen Infrastruktur  
Bsp.: Datacenter



### Probleme

- Gerätespezifische Anpassungen notwendig
- Unterschiedliche Schnittstellen
- Hohes Fehlerpotential
- Kenntnis für jedes Gerät erforderlich

### Lösung Part 1: YANG

→ Einheitliches Datenmodell

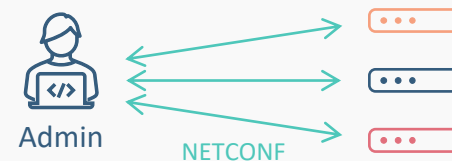


Use Case	Modell
Interfaces	ietf-interfaces
DHCP	ietf-dhcp



### Lösung Part 2: NETCONF

→ Einheitliche Schnittstelle



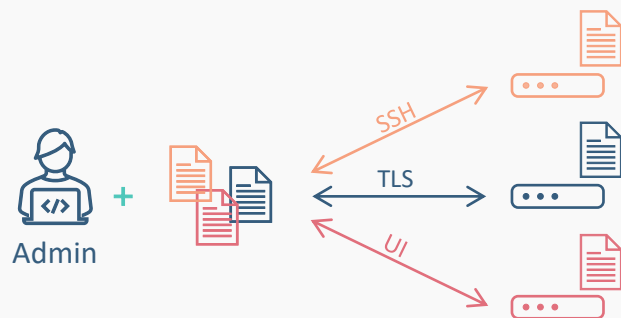
# MODELLBASIERTES MANAGEMENT

## PROBLEMSTELLUNG

MODELLBASIERTES MANAGEMENT | IETF DRAFT TCP | PROTOTYP - CLIXON | PORTIERUNG - QNX | FAZIT

### Ausgangssituation

Konfiguration einer heterogenen Infrastruktur  
Bsp.: Datacenter



### Probleme

- Gerätespezifische Anpassungen notwendig
- Unterschiedliche Schnittstellen
- Hohes Fehlerpotential
- Kenntnis für jedes Gerät erforderlich

### Lösung Part 1: YANG

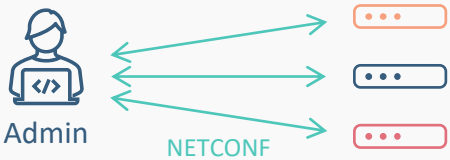
→ **Einheitliches Datenmodell**



Use Case	Modell
Interfaces	ietf-interfaces
DHCP	ietf-dhcp

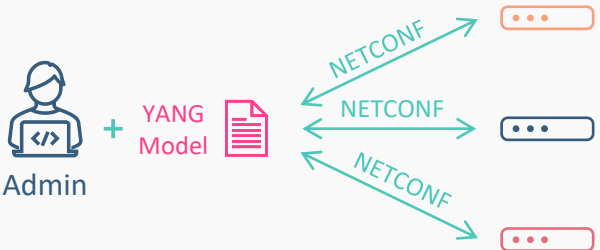
### Lösung Part 2: NETCONF

→ **Einheitliche Schnittstelle**



### Lösungsansatz

Modellbasiertes Management einer heterogenen Infrastruktur mit **YANG & NETCONF**

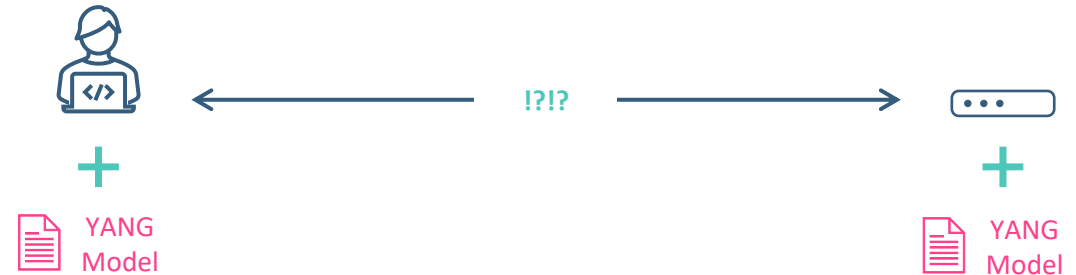


### Lösung

- Einheitliches Datenmodell → YANG Modelle
  - Einheitliche Schnittstelle → NETCONF
- **erleichtert Automatisierung auch bei heterogener Infrastruktur**

### | Lösung Part 1: YANG

- Modulare Datenmodellierungssprache
- Einheitliche + standardisierte Datenmodelle
- Zustands- und Konfigurationsdaten
- Protokollunabhängig



# MODELLBASIERTES MANAGEMENT

YANG (Yet Another Next Generation)

MODELLBASIERTES MANAGEMENT | IETF DRAFT TCP | PROTOTYP - CLIXON | PORTIERUNG - QNX | FAZIT

Module

```
module ietf-tcp@2020-11-16 {
```

```
}
```

# MODELLBASIERTES MANAGEMENT

YANG (Yet Another Next Generation)

MODELLBASIERTES MANAGEMENT | IETF DRAFT TCP | PROTOTYP - CLIXON | PORTIERUNG - QNX | FAZIT

Module

Metadaten

```
module ietf-tcp@2020-11-16 {  
  yang-version "1.1";  
  namespace "urn:ietf:params:xml:ns:yang:ietf-tcp";  
  prefix "tcp";  
  revision "2020-11-16" {  
    description "Initial Version";  
  }  
}
```



# MODELLBASIERTES MANAGEMENT

YANG (Yet Another Next Generation)

MODELLBASIERTES MANAGEMENT | IETF DRAFT TCP | PROTOTYP - CLIXON | PORTIERUNG - QNX | FAZIT

Module

Metadaten

Import Modules

```
module ietf-tcp@2020-11-16 {  
  yang-version "1.1";  
  namespace "urn:ietf:params:xml:ns:yang:ietf-tcp";  
  prefix "tcp";  
  revision "2020-11-16" {  
    description "Initial Version";  
  }  
  import ietf-inet-types {  
    prefix "inet";  
  }  
}
```

# MODELLBASIERTES MANAGEMENT

YANG (Yet Another Next Generation)

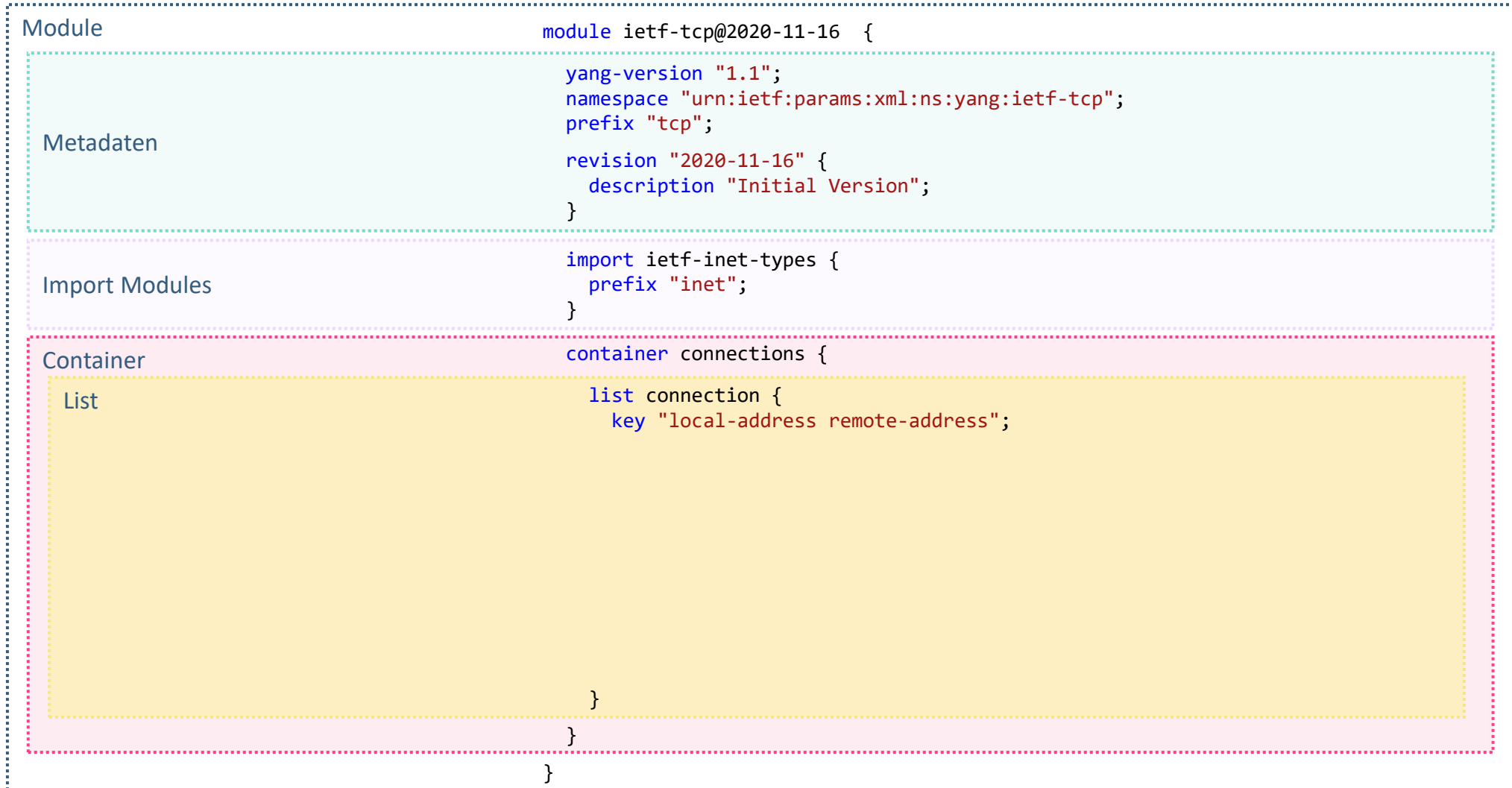
MODELLBASIERTES MANAGEMENT | IETF DRAFT TCP | PROTOTYP - CLIXON | PORTIERUNG - QNX | FAZIT



# MODELLBASIERTES MANAGEMENT

YANG (Yet Another Next Generation)

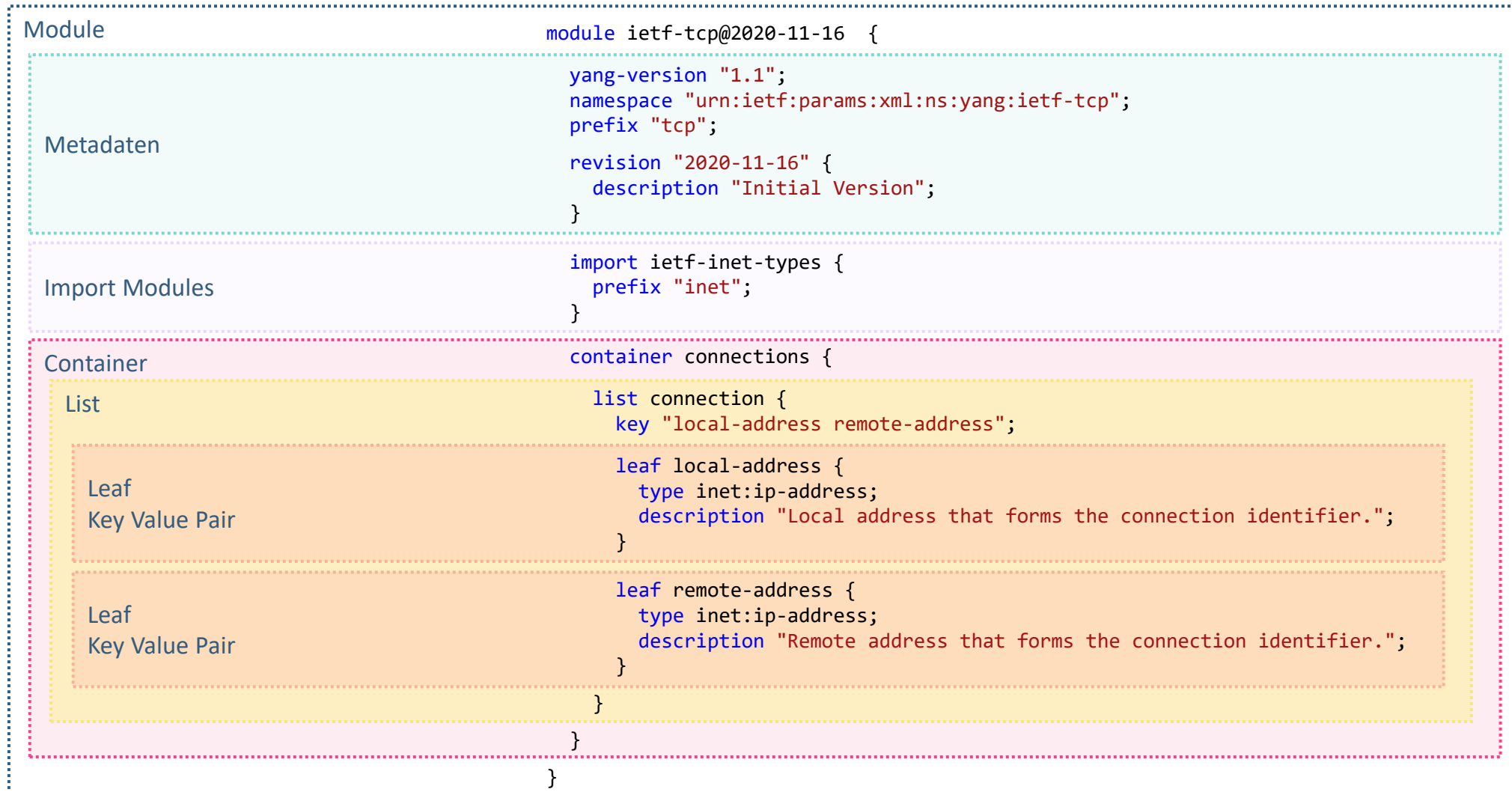
MODELLBASIERTES MANAGEMENT | IETF DRAFT TCP | PROTOTYP - CLIXON | PORTIERUNG - QNX | FAZIT



# MODELLBASIERTES MANAGEMENT

YANG (Yet Another Next Generation)

MODELLBASIERTES MANAGEMENT | IETF DRAFT TCP | PROTOTYP - CLIXON | PORTIERUNG - QNX | FAZIT



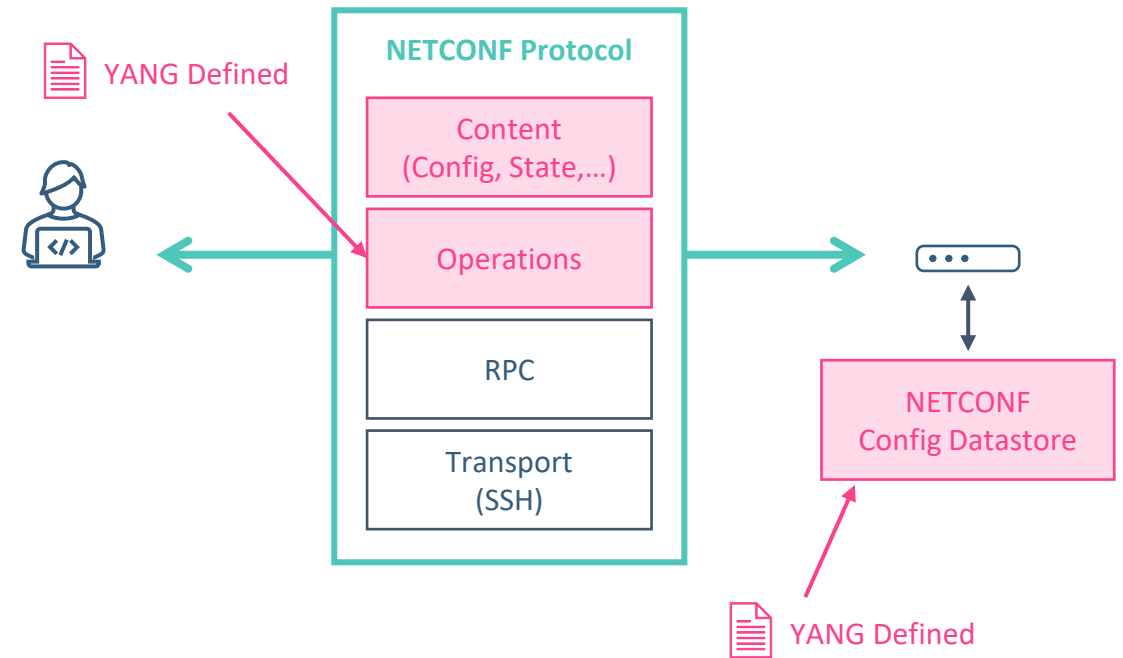
### | Lösung Part 1: YANG

- Modulare Datenmodellierungssprache
- Einheitliche + standardisierte Datenmodelle
- Zustands- und Konfigurationsdaten
- Protokollunabhängig



### | Lösung Part 2: NETCONF

- Netzwerkmanagementprotokoll
- Zustands- und Konfigurationsdaten lesen und ggf. modifizieren
- XML-kodierte Nachrichten als RPC bspw. über SSH



## IETF DRAFT TCP

---



Application Layer
Transport Layer
Internet Layer
Link Layer



RFC Entwurf der TCPM Working Group (IETF)

Zustands- und Konfigurationsdaten des TCP/IP-Stacks

Globale Sicht auf TCP-Verbindungen aus Sicht des Betriebssystems

Mitarbeit der HS Esslingen unter der Leitung von Prof. Scharf

### | draft-ietf-tcpm-yang-tcp

```
module: ietf-tcp
  +--rw tcp!
```



TCP Container

```
    +--rw connections
        ...
```



Liste mit TCP-Verbindungen  
local-address, remote-address, local-port, remote-port

```
    +--rw server {server}?
        ...
```



TCP Server Konfiguration (Passive Verbindungen)

```
    +--rw client {client}?
        ...
```



TCP Client Konfiguration (Aktive Verbindungen)

```
    +--ro statistics {statistics}?
        ...
```



Statistik (config false)  
attempt-fails, currently-established, in/out-segments, ...



# Prototyp

---

### | 1. Einarbeitungsphase

- Einarbeitung in relevante RFCs
- Anforderungen an System- und Testumgebung



### | 2. Marktuntersuchung

- Analyse bestehender NETCONF / YANG Frameworks
- **Fokus auf:**
  - Portierbarkeit
  - OpenSource
  - Wenig Abhängigkeiten / Systemanforderungen



### | 3. Prototypische Implementierung

- Clixon Best Fit für Use Case
- Wenig Abhängigkeiten
- Hohe Testabdeckung + Codequalität

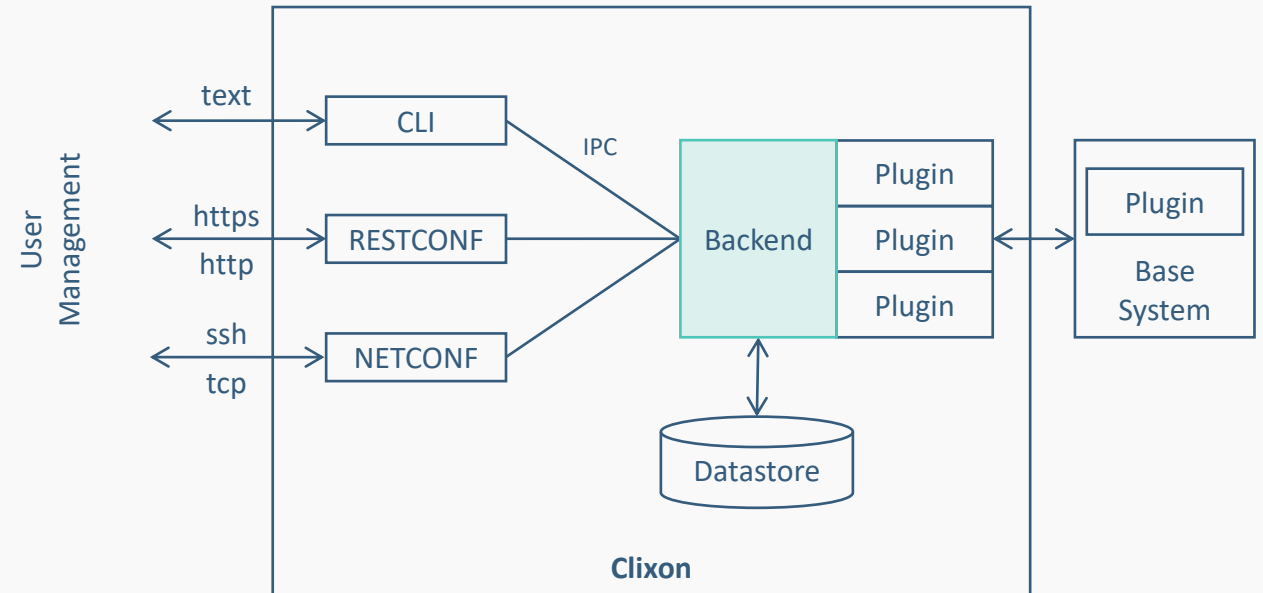
**“Clixon is a YANG-based configuration manager, with interactive CLI, NETCONF and RESTCONF interfaces, an embedded database and transaction mechanism.”**

- Clixon Github

### Backend

- Core-Funktionalität
- Privilege Management

### Systemarchitektur



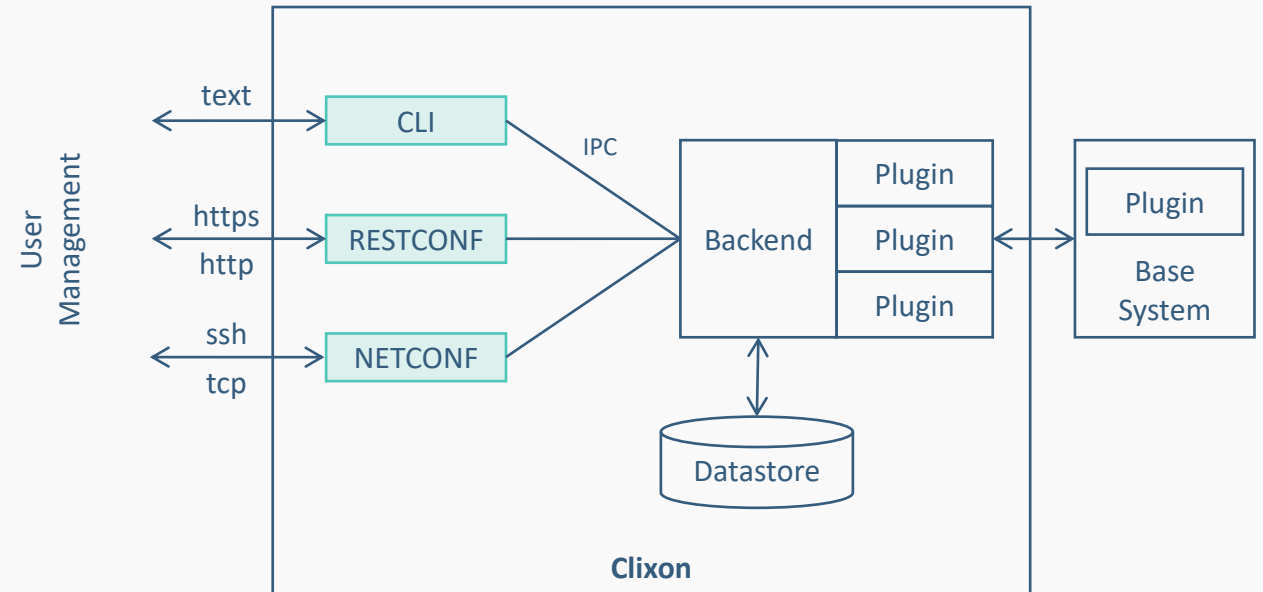
### Backend

- Core-Funktionalität
- Privilege Management

### CLI, RESTCONF, NETCONF

- Separate Prozesse
- Inter-process communication (IPC) mit Backend

### Systemarchitektur



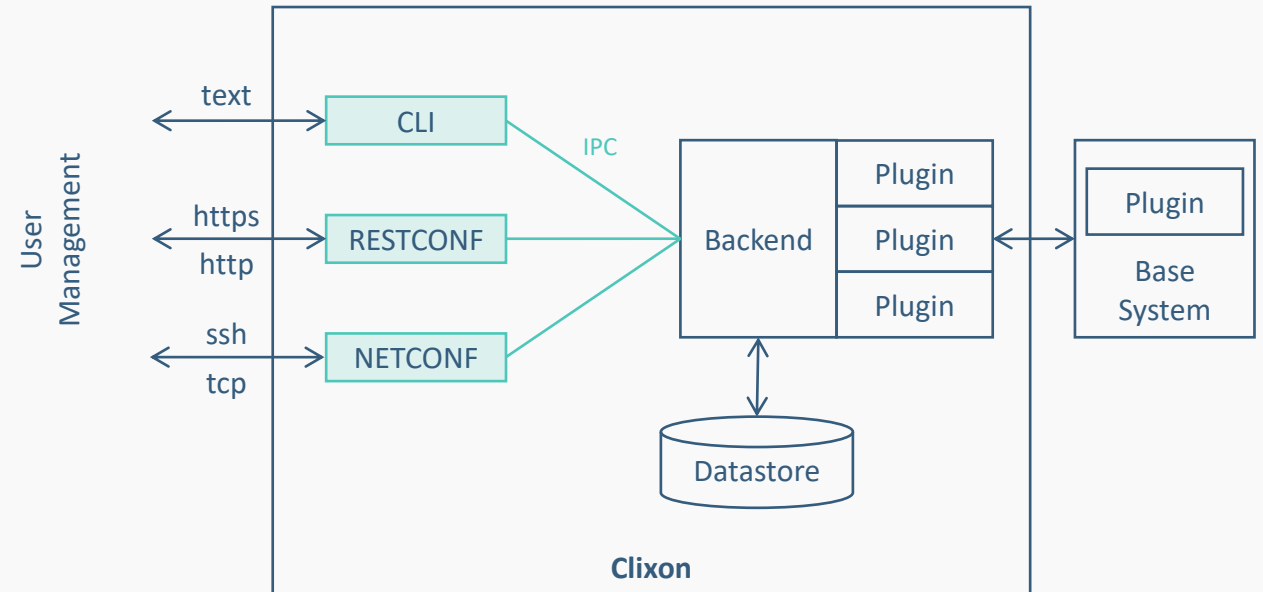
### Backend

- Core-Funktionalität
- Privilege Management

### CLI, RESTCONF, NETCONF

- Separate Prozesse
- Inter-process communication (IPC) mit Backend

### Systemarchitektur



### Backend

- Core-Funktionalität
- Privilege Management

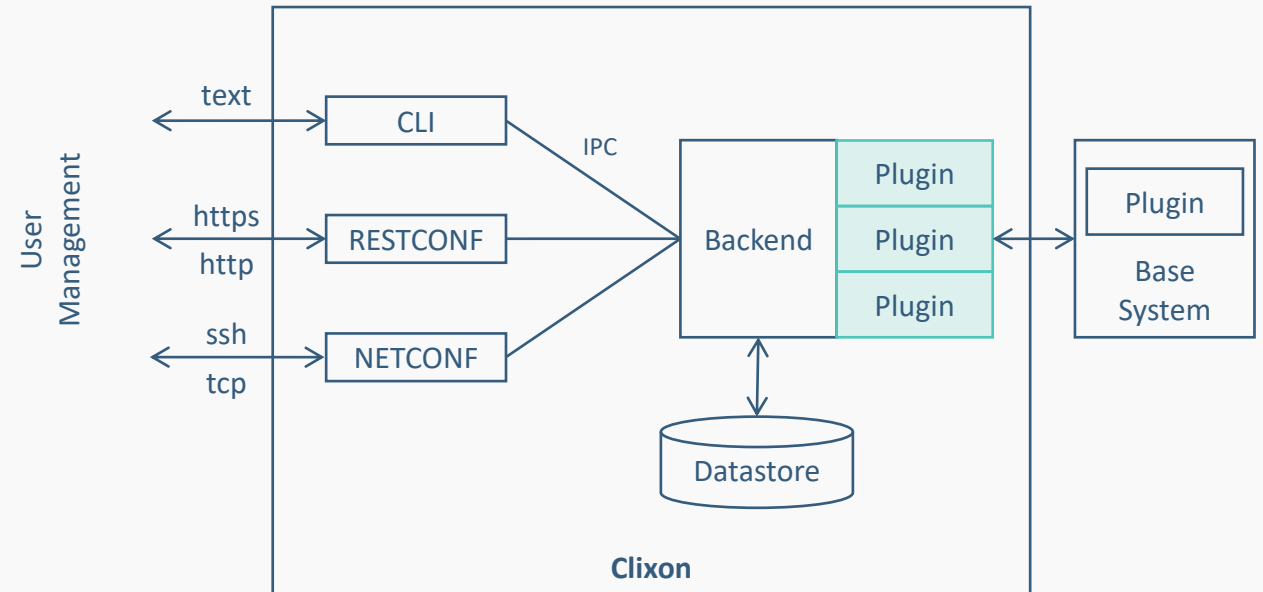
### CLI, RESTCONF, NETCONF

- Separate Prozesse
- Inter-process communication (IPC) mit Backend

### Plugins

- Dynamisch geladene Shared-Objects
- Registrierung von Callbacks
- Alternative Architektur: Plugin außerhalb von Clixon

### Systemarchitektur



### Backend

- Core-Funktionalität
- Privilege Management

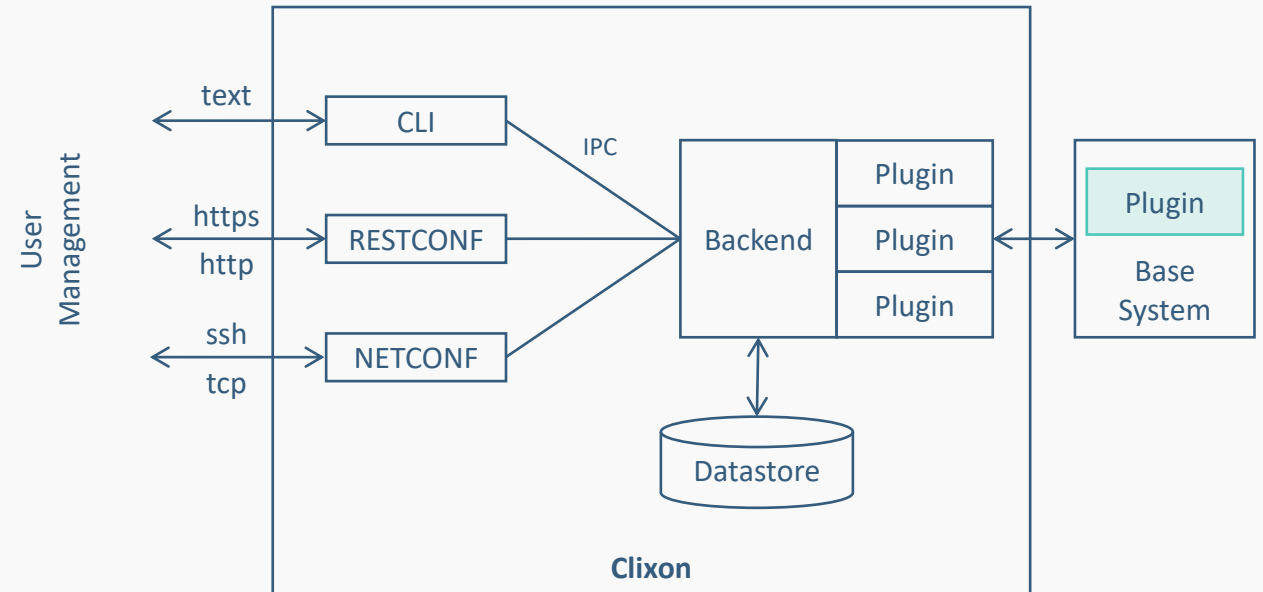
### CLI, RESTCONF, NETCONF

- Separate Prozesse
- Inter-process communication (IPC) mit Backend

### Plugins

- Dynamisch geladene Shared-Objects
- Registrierung von Callbacks
- Alternative Architektur: Plugin außerhalb von Clixon

### Systemarchitektur





### Backend

- Core-Funktionalität
- Privilege Management

### CLI, RESTCONF, NETCONF

- Separate Prozesse
- Inter-process communication (IPC) mit Backend

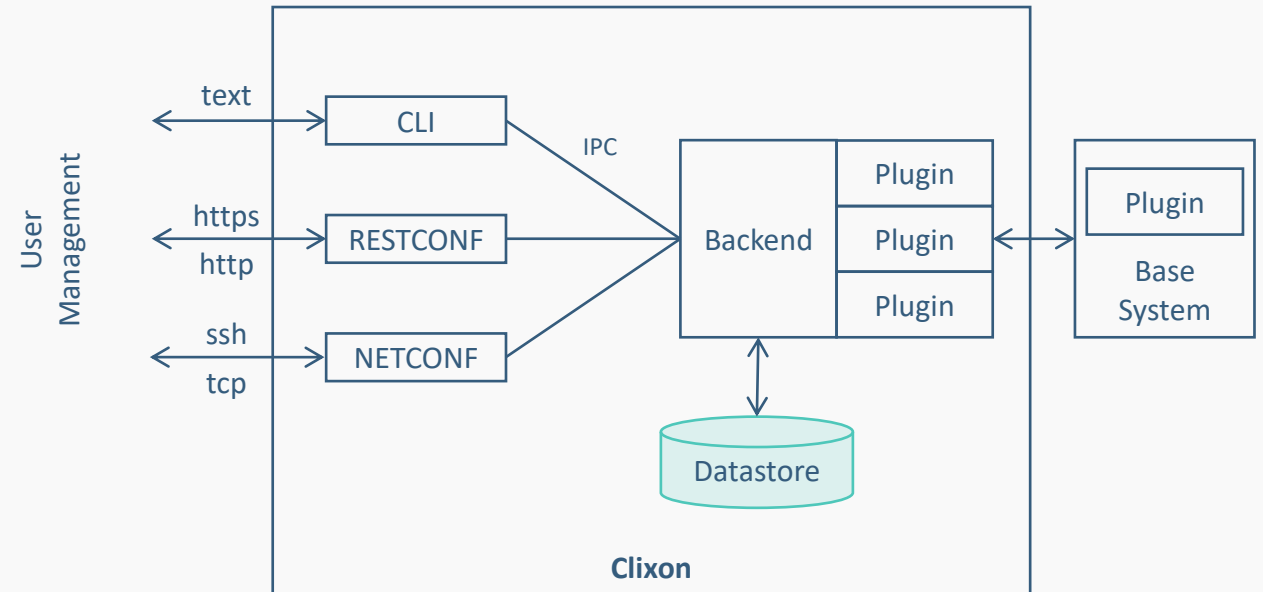
### Plugins

- Dynamisch geladene Shared-Objects
- Registrierung von Callbacks
- Alternative Architektur: Plugin außerhalb von Clixon

### Datastores

- Speichert die Konfigurationsdaten des Systems
- Zugriff via Clixon-API

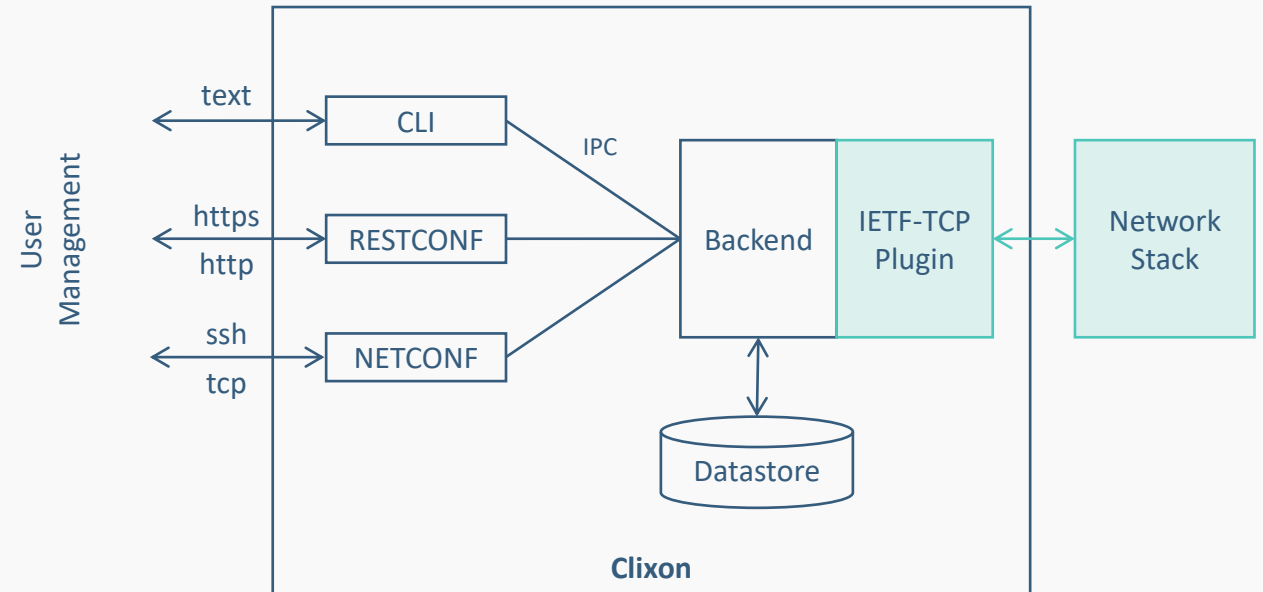
### Systemarchitektur



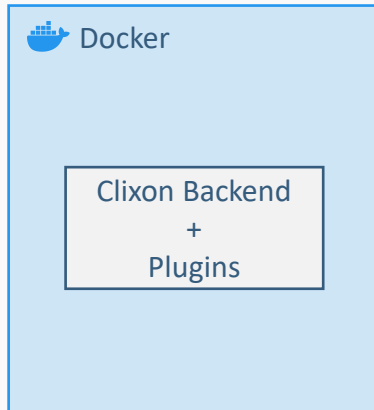
### TCP Plugin

- Läuft als Teil des Clixon-Backend-Prozess
- Plugin implementiert Schnittstelle zu Network-Stack
- Buildsystem: CMake
- Programmiersprache: C++

### Systemarchitektur



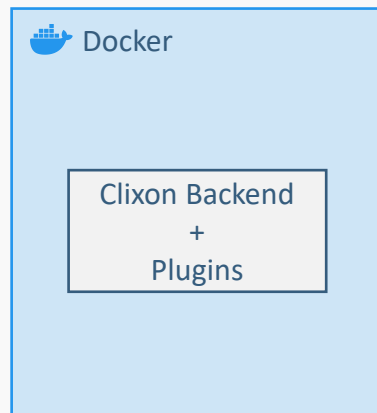
### | Docker Container



### | Vorteile

- Reproduzierbare Entwicklungsumgebung
- Einfaches Deployment

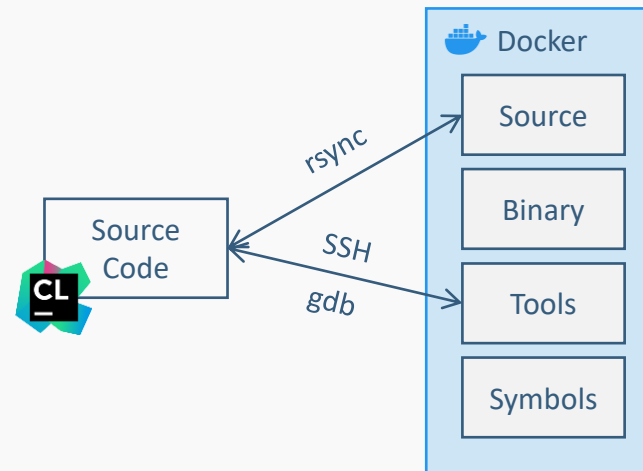
### | Docker Container



#### | Vorteile

- Reproduzierbare Entwicklungsumgebung
- Einfaches Deployment

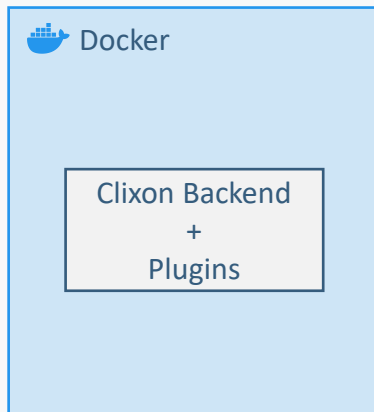
### | CLion Remote Development



#### | Vorteile

- Entwicklung auf Windows Host mit Linux Target
- Remote Debugging

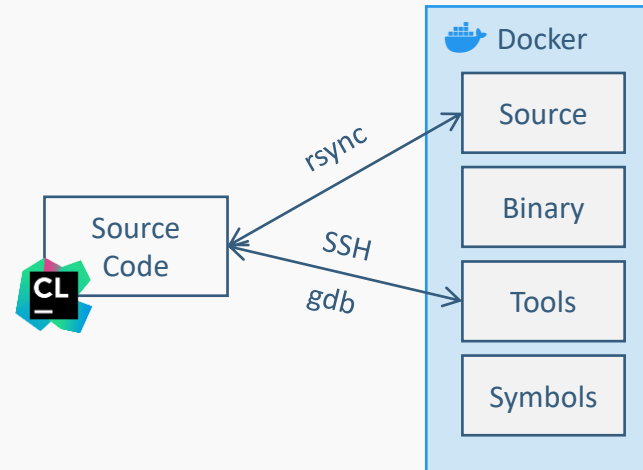
### Docker Container



#### Vorteile

- Reproduzierbare Entwicklungsumgebung
- Einfaches Deployment

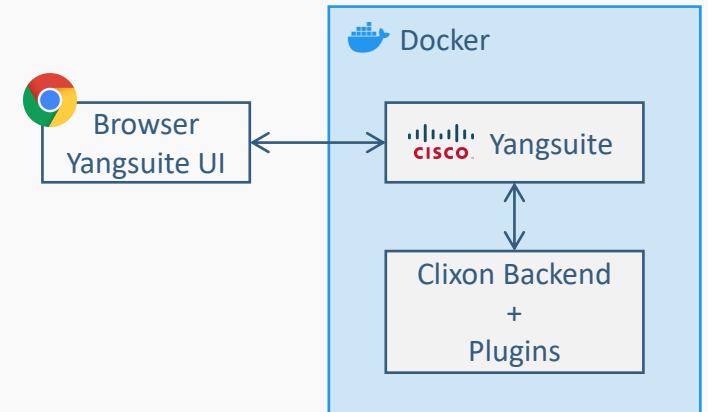
### CLion Remote Development



#### Vorteile

- Entwicklung auf Windows Host mit Linux Target
- Remote Debugging

### Yangsuite



#### Vorteile

- Übersichtliches UI
- Optimal für manuelles Testen der Plugins

Framing Protocol (bisher)

```

<?xml version="1.0" encoding="UTF-8"?>
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <capabilities>
    <capability>
      urn:ietf:params:netconf:base:1.1
    </capability>
  </capabilities>
</hello>
]]>]]> -> Message Separator Sequence

```



Chunked Framing Mechanism (neu -> RFC 6242 Section 4.2)

```

\n#4\n
<rpc
\n#18\n
  message-id="102"\n
\n#79\n
    xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">\n
  <close-session/>\n
</rpc>
\n##\n

```



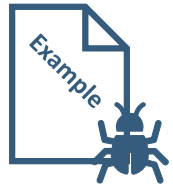
Pull Request

clicon / clixon

<> Code Issues 20 Pull requests 2 Actions Projects Wiki Security Insights

Netconf Chunks (RFC6242 - Section 4.2) #217

Draft s-bauer wants to merge 15 commits into clicon:master from s-bauer:feature/netconf-chunks



### | Bug in Example Code

<https://github.com/clicon/clixon/pull/161>



### | XML Namespace Issue

<https://github.com/clicon/clixon/pull/162>



### | Build Script Improvements

<https://github.com/clicon/clixon/pull/197>



### | Memory Corruption

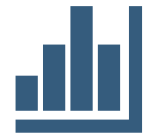
<https://github.com/clicon/clixon/pull/238>



| Schreibzugriff auf TCP-Verbindungsliste



| Client/Server Imports



| Zurücksetzen TCP Statistiken



## | draft-ietf-tcpm-yang-tcp

```
module: ietf-tcp
+--rw tcp!
```



TCP Container

```
+--rw connections
...
```



Liste mit TCP-Verbindungen  
local-address, remote-address, local-port, remote-port

```
+--rw server {server}?
...
```



TCP Server Konfiguration (Passive Verbindungen)

```
+--rw client {client}?
...
```



TCP Client Konfiguration (Aktiven Verbindungen)

```
+--ro statistics {statistics}?
...
```



Statistik (config false)  
attempt-fails, currently-established, in/out-segments, ...

## Portierung - QNX

---

### | Zwischenstand

- TCP Plugin für Ubuntu OS lauffähig
- Nicht für Ubuntu gedacht sondern Embedded Geräte (Switches, Router, AP, Firewalls, ...)

### | Einsatzgebiete für NETCONF / YANG

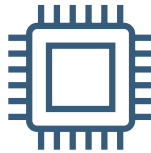
- Modellbasiertes Management auch in anderen Branchen relevant bspw. Medical, Robotics, Automotive
- Anderen Anforderungen -> Real Time OS bspw. QNX
  - | **Microkernel Reliability**
  - | **Real-Time Availability**
  - | **Comprehensive, Layered Security**

### | Portierung auf QNX

- Portierung von Clixon inkl. TCP Plugin auf QNX RTOS



| Bare Minimum



| Näher an der Hardware (Netzwerkstack,  
Explizite Aktionen)

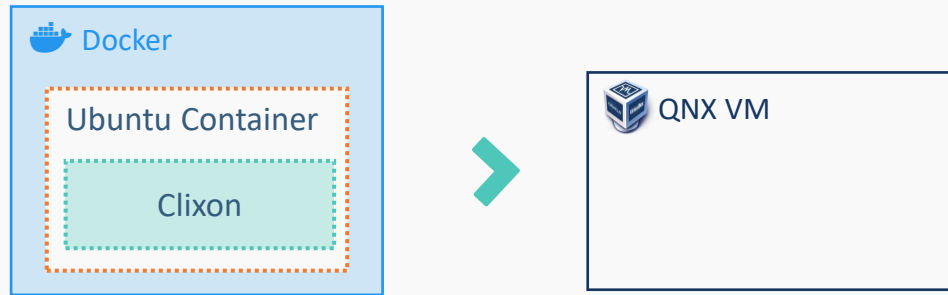


| Eingeschränktes Programmierinterface  
(im Vergleich zu UNIX/Linux)

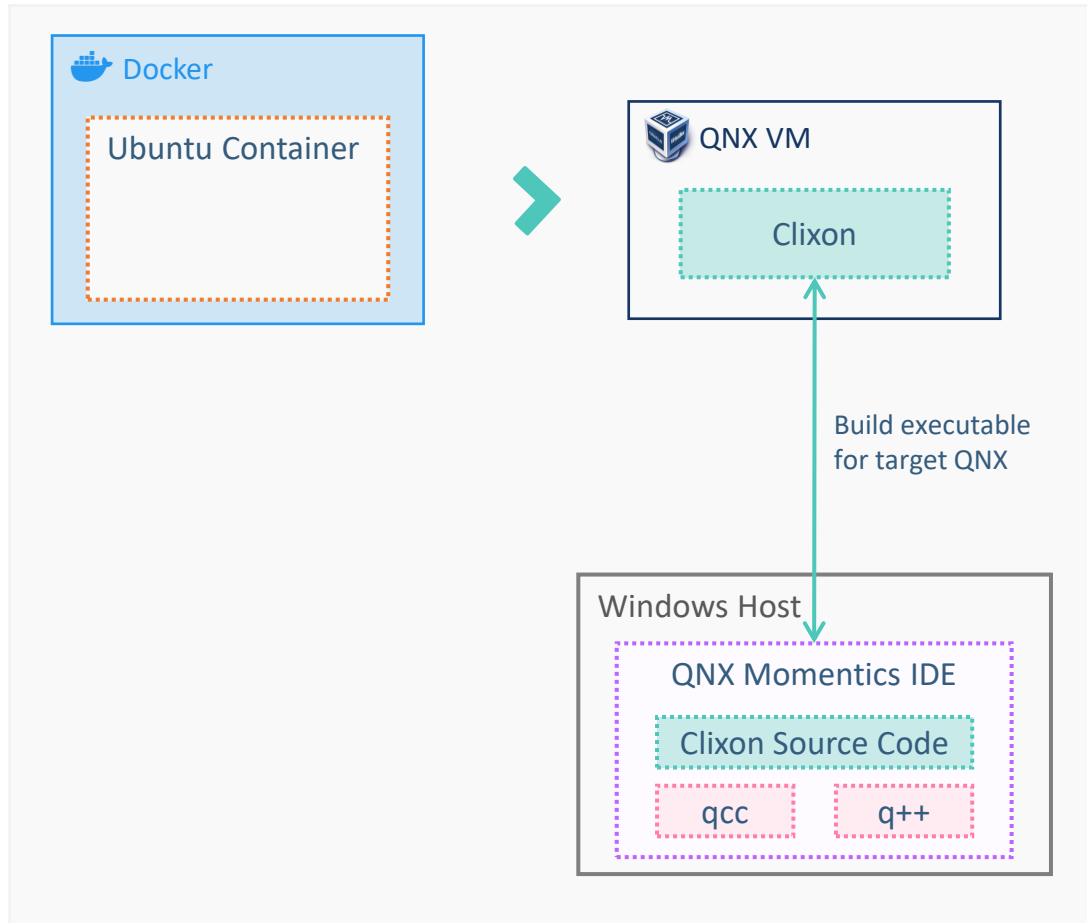
**POSIX**

| POSIX-Konform

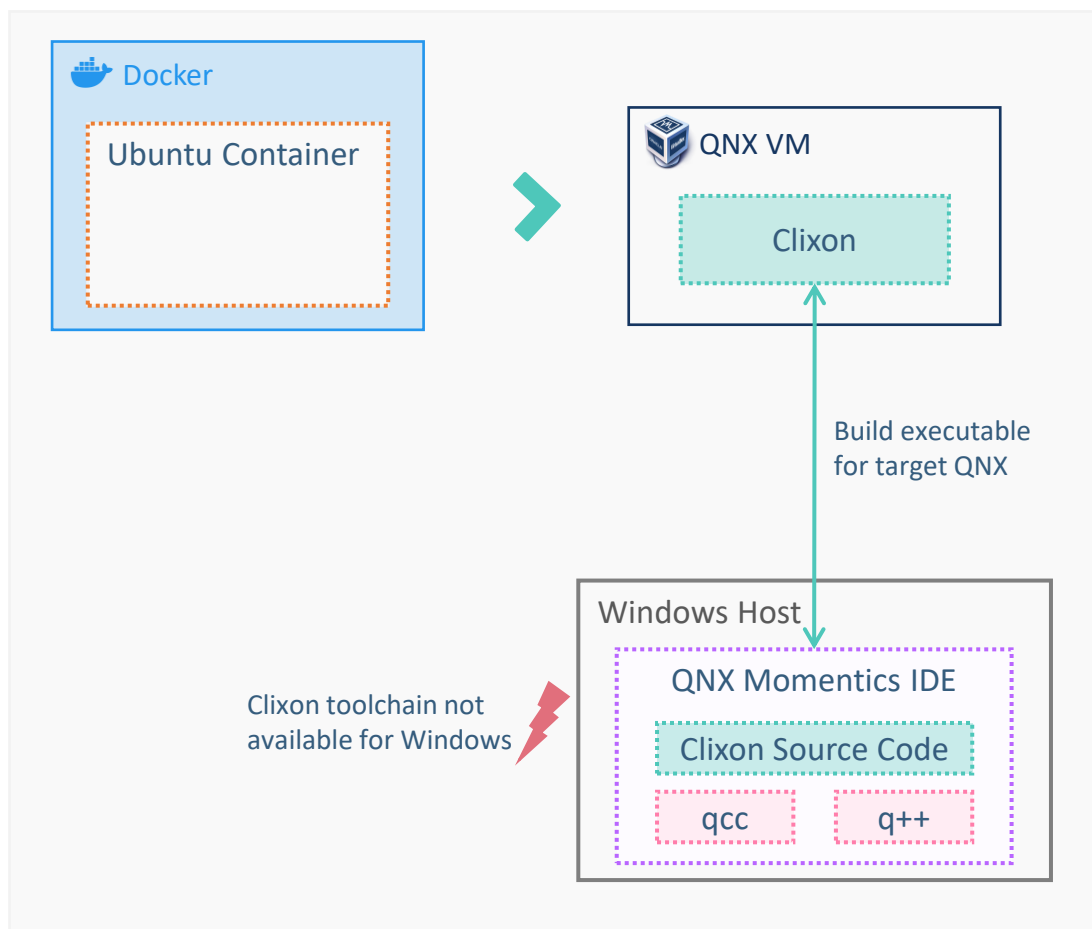
### 1. Portierung Ubuntu Container -> QNX Neutrino VM



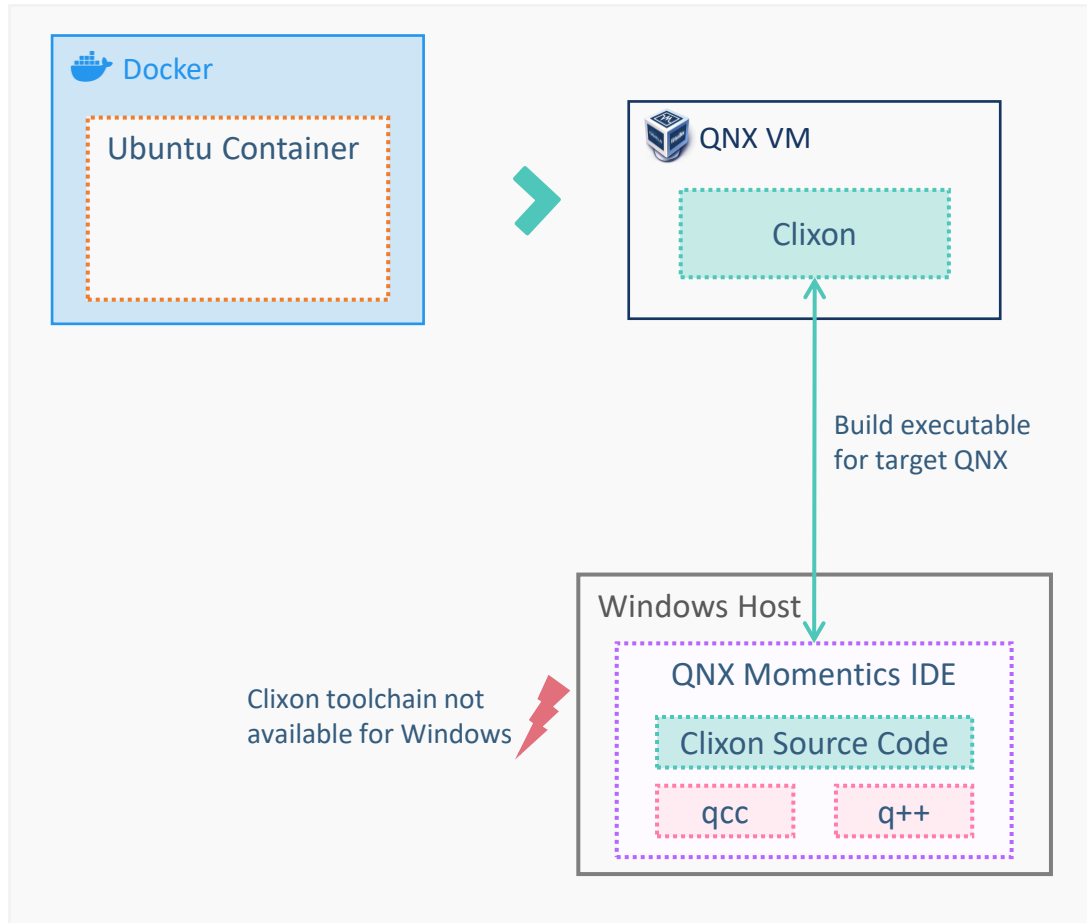
### 1. Portierung Ubuntu Container -> QNX Neutrino VM



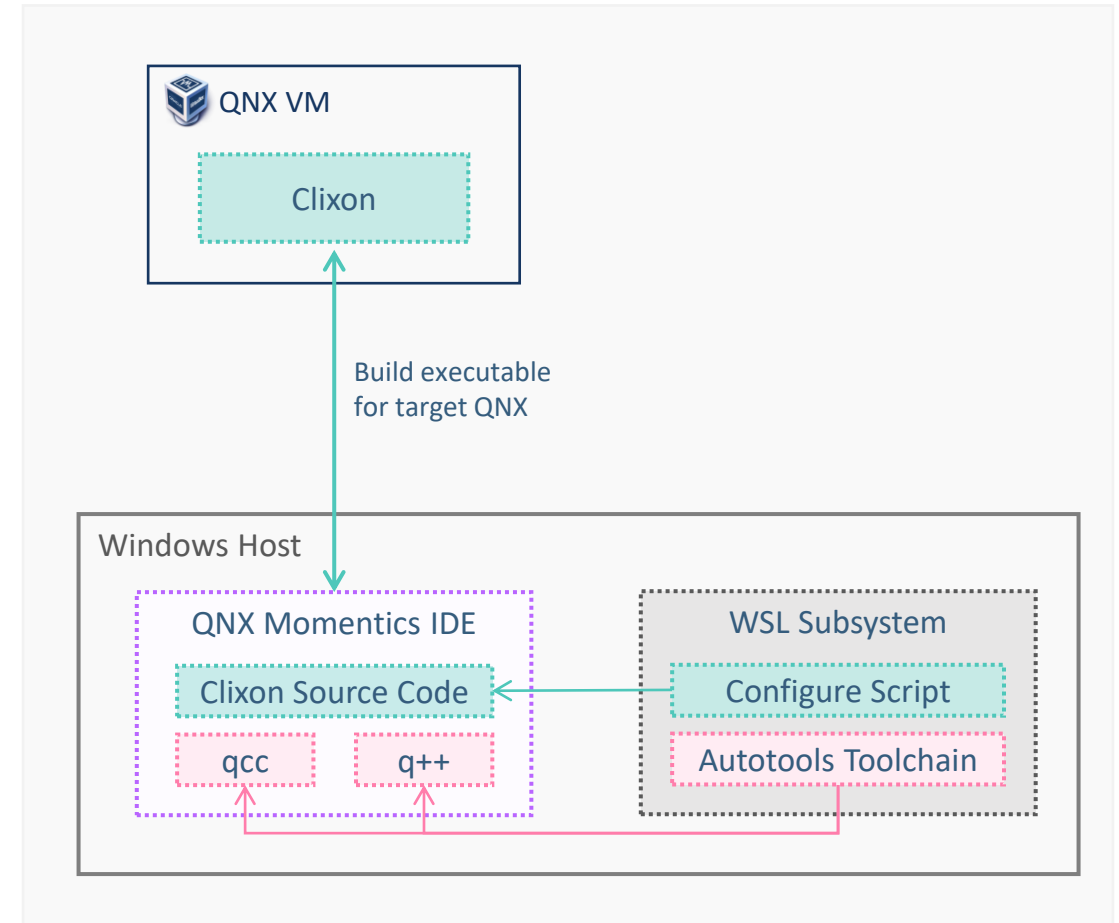
### 1. Portierung Ubuntu Container -> QNX Neutrino VM



### 1. Portierung Ubuntu Container -> QNX Neutrino VM



### 2. WSL Subsystem für Autotools Toolchain





### | struct dirent - Memory Corruption

- Speicherkorruption beim Lesen der Yang-Modelle von der Festplatte
- POSIX-Dokumentation von `struct dirent`

```
|  ino_t    d_ino      File serial number.  
|  char    d_name[]   Name of entry.
```

- Besonderheit

```
|  The array of char d_name is not a fixed size.
```



### | Lösung

- Fixed: Memory Allocation -> Pull Request

 [clixon](#) / [clixon](#)

[Code](#) [Issues](#) 20 [Pull requests](#) 2 [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#)

Fixed memory allocation for `struct dirent` #238

 **Merged** olofhagsand merged 1 commit into [clixon:master](#) from [mager-m:master](#)  on Jun 11

### | struct dirent - Memory Corruption

- Speicherkorruption beim Lesen der Yang-Modelle von der Festplatte
- POSIX-Dokumentation von struct dirent

```
|  ino_t    d_ino      File serial number.  
|  char    d_name[]   Name of entry.
```

- Besonderheit

```
|  The array of char d_name is not a fixed size.
```

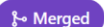
### | Lösung

- Fixed: Memory Allocation -> Pull Request

 clicon / clixon

[Code](#) [Issues 20](#) [Pull requests 2](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#)

Fixed memory allocation for struct dirent #238

 Merged olofhagsand merged 1 commit into clicon:master from mager-m:master on Jun 11

### | setresuid

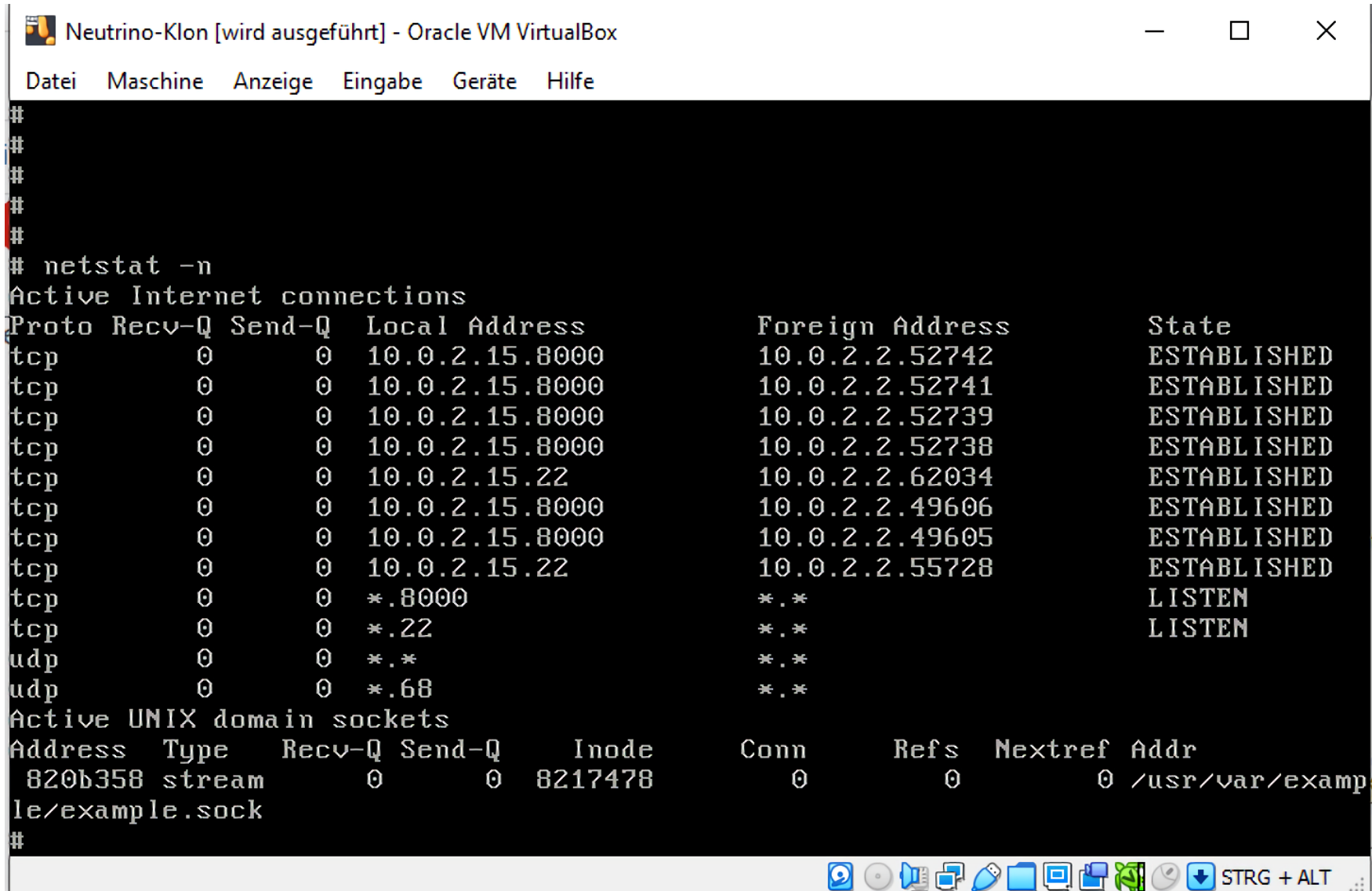
- setresuid nicht auf QNX Neutrino unterstützt
- setresuid jetzt über Featureflag de/aktivierbar

### | Include header

- Header teilweise unter anderen Verzeichnisstrukturen
- Bsp.: statt <time.h> → <sys/time.h>

### | /proc/net Verzeichnis

- /proc/net-Verzeichnis auf QNX Neutrino nicht verfügbar
- Entweder via Sys-Calls oder netstat output parsen



```
#
#
#
#
# netstat -n
Active Internet connections
Proto Recv-Q Send-Q Local Address Foreign Address State
tcp 0 0 10.0.2.15.8000 10.0.2.2.52742 ESTABLISHED
tcp 0 0 10.0.2.15.8000 10.0.2.2.52741 ESTABLISHED
tcp 0 0 10.0.2.15.8000 10.0.2.2.52739 ESTABLISHED
tcp 0 0 10.0.2.15.8000 10.0.2.2.52738 ESTABLISHED
tcp 0 0 10.0.2.15.22 10.0.2.2.62034 ESTABLISHED
tcp 0 0 10.0.2.15.8000 10.0.2.2.49606 ESTABLISHED
tcp 0 0 10.0.2.15.8000 10.0.2.2.49605 ESTABLISHED
tcp 0 0 10.0.2.15.22 10.0.2.2.55728 ESTABLISHED
tcp 0 0 *.8000 *.* LISTEN
tcp 0 0 *.22 *.* LISTEN
udp 0 0 *.* *.*
udp 0 0 *.68 *.*
Active UNIX domain sockets
Address Type Recv-Q Send-Q Inode Conn Refs Nextref Addr
820b358 stream 0 0 8217478 0 0 0 /usr/var/example/example.sock
#
```

Cisco YANG Suite

Admin

Setup

Analytics

Explore

Protocols

Help

YANG Suite / NETCONF / YANG set "tcp-set" / Modules

NETCONF

sim

YANG Settcp-set

Module(s)ietf-tcp x

Load Module(s)

NETCONF Operationget

Deviceclixon\_qnx

Edit Device

Open Device Window

YANG Tree

Replays

RPC Options...

Build RPC

Run RPC(s)

Clear RPC(s)

Nodes

Value

ietf-tcp

tcp

connections

connection

local-address

remote-address

local-port

remote-port

common

keepalives

idle-time

max-probes

probe-interval

authentication

ao

md5

statistics

active-opens

passive-opens

attempt-fails

establish-resets

currently-established

in-segments

out-segments

retransmitted-segments

in-errors

out-resets

reset

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <get>
    <filter>
      <tcp xmlns="urn:ietf:params:xml:ns:yang:ietf-tcp">
        <connections>
          <connection/>
        </connections>
      </tcp>
    </filter>
  </get>
</rpc>
```

### Client Request

Sending:

```
#379
<nc:rpc xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="urn:uuid:00a55c7c-077c-4b2d-8ebf-b1bc64dd59ea"><nc:get-config>
  <nc:source>
    <nc:running/>
  </nc:source>
  <nc:filter>
    <tcp xmlns="urn:ietf:params:xml:ns:yang:ietf-tcp">
      <connections/>
    </tcp>
  </nc:filter>
</nc:get-config>
</nc:rpc>

##
```

### Server Response

Received message from host

```
<?xml version="1.0" ?>
<rpc-reply message-id="urn:uuid:00a55c7c-077c-4b2d-8ebf-b1bc64dd59ea" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
    <tcp xmlns="urn:ietf:params:xml:ns:yang:ietf-tcp">
      <connections>
        <connection>
          <local-address>10.0.2.15</local-address>
          <remote-address>10.0.2.2</remote-address>
          <local-port>22</local-port>
          <remote-port>55728</remote-port>
        </connection>
        <connection>
          <local-address>10.0.2.15</local-address>
          <remote-address>10.0.2.2</remote-address>
          <local-port>22</local-port>
          <remote-port>62034</remote-port>
        </connection>
        <connection>
          <local-address>10.0.2.15</local-address>
          <remote-address>10.0.2.2</remote-address>
          <local-port>8000</local-port>
          <remote-port>49605</remote-port>
        </connection>
      </connections>
    </tcp>
  </data>
</rpc-reply>
```

## Fazit

---

### | Step 1 IETF Draft

- Einarbeitung – viele RFCs gelesen etc.
- Mitwirkung an einem neuen Internetstandard
- Findings aus Prototyp ins Modell geflossen



### | Learnings

- Umfangreich, aber sehr gut verständlich
- Mitarbeit an (neuen) Internetstandards für jeden möglich
- Aktive Community

### | Step 1 IETF Draft

- Einarbeitung – viele RFCs gelesen etc.
- Mitwirkung an einem neuen Internetstandard
- Findings aus Prototyp ins Modell geflossen



### | Learnings

- Umfangreich, aber sehr gut verständlich
- Mitarbeit an (neuen) Internetstandards für jeden möglich
- Aktive Community

### | Step 2 Clixon

- Einarbeitung in umfangreiche Softwarelösung
- Prototypische Implementierung
- Contributing to OpenSource



### | Learnings

- OpenSource auch für Nischenprodukte
- Leichte Mitarbeit möglich
- Sind über jede Hilfe froh



### | Step 1 IETF Draft

- Einarbeitung – viele RFCs gelesen etc.
- Mitwirkung an einem neuen Internetstandard
- Findings aus Prototyp ins Modell geflossen



### | Step 2 Clixon

- Einarbeitung in umfangreiche Softwarelösung
- Prototypische Implementierung
- Contributing to OpenSource



### | Step 3 QNX

- Einarbeitung in QNX – viel Dokumentation
- Entwicklungsumgebung aufgesetzt
- Clixon auf QNX portiert
- TCP Plugin an QNX angepasst



### | Learnings

- Umfangreich, aber sehr gut verständlich
- Mitarbeit an (neuen) Internetstandards für jeden möglich
- Aktive Community

### | Learnings

- OpenSource auch für Nischenprodukte
- Leichte Mitarbeit möglich
- Sind über jede Hilfe froh

### | Learnings

- Bare Minimum OS
- POSIX-Standard erleichtert die Portierung
- Online wenig Hilfestellung → Dokumentation



HOCHSCHULE  
**ESSLINGEN**

## | Folie 5

In Anlehnung an A. Bierman, Network Configuration Protocol. [Online]. Verfügbar unter: [http://www.netconfcentral.org/netconf\\_docs](http://www.netconfcentral.org/netconf_docs). Zugriff am 28. Juli 2021

## | Folie 7

Prof. Scharf Porträt: Hochschule Esslingen. [Online]. Verfügbar unter: <https://www.hs-esslingen.de/personen/michael-scharf/>. Zugriff am 28. Juli 2021

## | Folie 12 + 13

In Anlehnung an: O. Hagsand, clixon-docs. [Online]. Verfügbar unter: <https://clixon-docs.readthedocs.io/en/latest/overview.html#system-architecture>. Zugriff am 28. Juli 2021

## | Logos:

- IETF: IETF. [Online]. Verfügbar unter: [https://upload.wikimedia.org/wikipedia/commons/9/98/IETF\\_Logo.svg](https://upload.wikimedia.org/wikipedia/commons/9/98/IETF_Logo.svg). Zugriff am 28. Juli 2021
- Hochschule Esslingen: Hochschule Esslingen. [Online]. Verfügbar unter: <https://www.hs-esslingen.de/>. Zugriff am 28. Juli 2021
- docker: docker, Docker Logos. [Online]. Verfügbar unter: <https://www.docker.com/company/newsroom/media-resources>. Zugriff am 28. Juli 2021
- CLion: JetBrains, CLion. [Online]. Verfügbar unter: <https://www.jetbrains.com/company/brand/logos/>. Zugriff am 28. Juli 2021. Copyright © 2021 JetBrains s.r.o. CLion and the CLion Logo are registered trademarks of JetBrains s.r.o.
- Chrome Logo: Google, Google Chrome. [Online]. Verfügbar unter: [https://upload.wikimedia.org/wikipedia/commons/a/a5/Google\\_Chrome\\_icon\\_%28September\\_2014%29.svg](https://upload.wikimedia.org/wikipedia/commons/a/a5/Google_Chrome_icon_%28September_2014%29.svg). Zugriff am 28. Juli 2021
- Cisco Logo: Cisco, cisco-logo. [Online]. Verfügbar unter: [https://upload.wikimedia.org/wikipedia/commons/6/64/Cisco\\_logo.svg](https://upload.wikimedia.org/wikipedia/commons/6/64/Cisco_logo.svg). Zugriff am 28. Juli 2021
- VirtualBox: Oracle Corporation, VirtualBox 4.2. for Windows hosts. [Online]. Verfügbar unter: [https://upload.wikimedia.org/wikipedia/commons/d/d5/Virtualbox\\_logo.png](https://upload.wikimedia.org/wikipedia/commons/d/d5/Virtualbox_logo.png). Zugriff am 28. Juli 2021
- Clixon: O. Hagsand, Clixon. [Online]. Verfügbar unter: <https://clixon-docs.readthedocs.io/en/latest/>. Zugriff am 28. Juli 2021