Automation Clixon, SNC

NORDUnet Conference 2024 Bergen

NORDUnet
Nordic Gateway for Research & Education

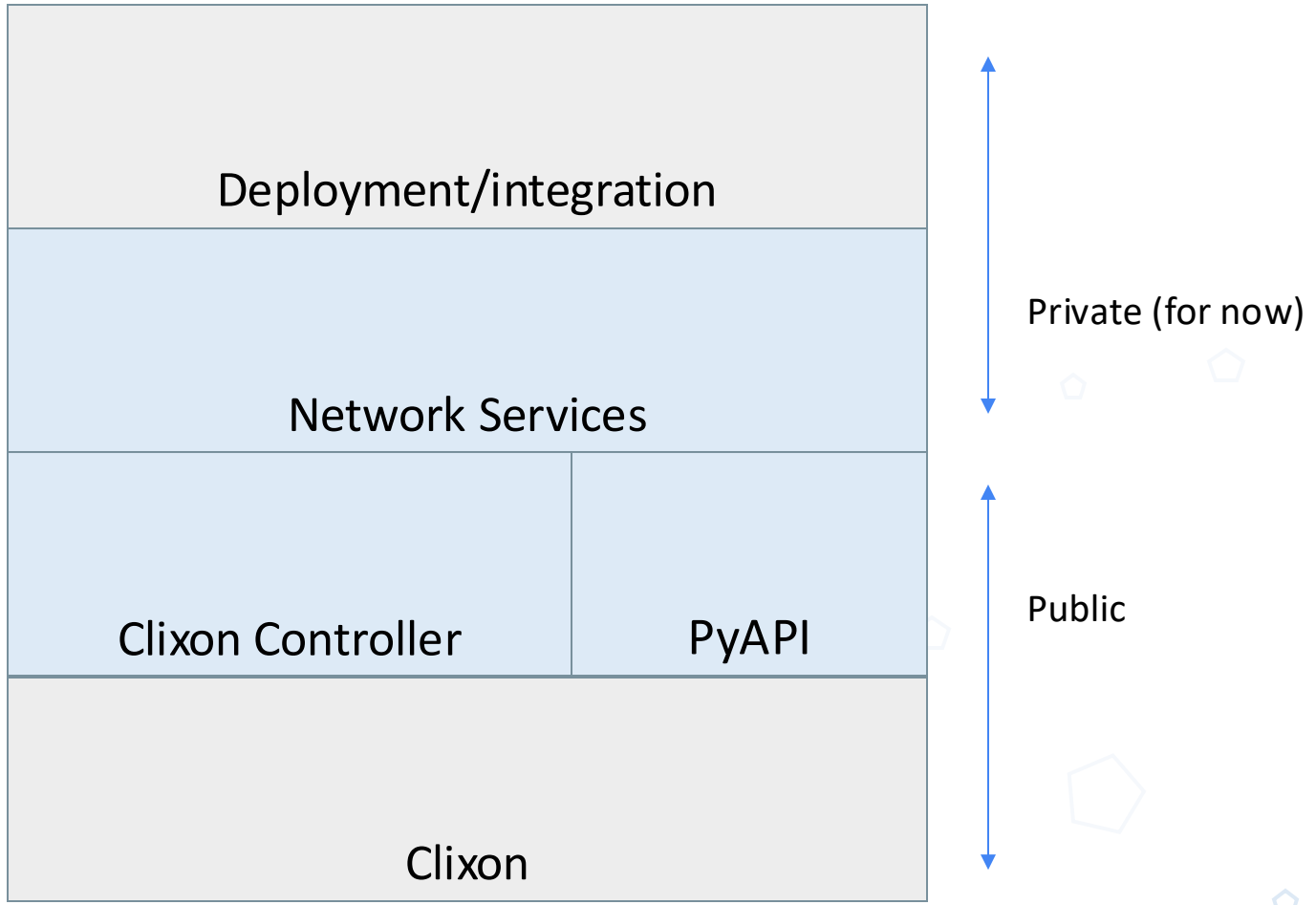
Markus Krogh

Background

- Joint project between SUNET and NORDUnet
- Development started December 2022
  - Based on existing open source “Clixon” project
- Based on NETCONF and YANG
  - Any device implementing RFC6241 + RFC 7950 “should” work
- Programmable services via a Python API
  - We developed services to handle BGP, user administration etc
- Support for multiple devices with different YANG schemas
  - We have a mix of Juniper PTX, MX, QFX etc
  - Possible to add other vendors too
- SUNET successfully did their 400G roll-out in Q1 2024

NORDUnet
Nordic Gateway for Research & Education

SNC structure



Clixon Controller

- Interactive CLI
  - **Based off of YANG models**
- Supports NETCONF, RESTCONF, SNMP
- Datastore of device and controller configuration (XML)
- Multi-vendor, multi-model devices
- Device push/pull
- Commit/transaction semantics across device-groups
  - **Edit, validate, commit**

Network Services

- Code with that makes configuration across multiple devices
- Written in Python, configured in CLI
- Current Services ~8 services
  - IBGP
  - BGP peering
  - AS-path filters
  - Prefix filters
  - SSH users / key distribution
  - IPCLOS
  - EVPN
  - l2vpn

Python service workflow

- Sync pull from devices
- Clixon CLI to create or change service
- Commit diff / commit trigger
  - When the services are committed the Python code is executed
- Python service manipulates device configuration tree
- Device config diffed against known state
- Device diff pushed to devices

File: ssh_users.py

```python
from clixon.element import Element
from clixon.helpers import get_devices_from_group, get_path, get_value

SERVICE = "ssh-users"


def setup(root, log, **kwargs):
    try:
        _ = root.services.ssh_users
    except Exception:
        return

    service_name = SERVICE
    instance_name = kwargs["instance"]

    attributes = {
        "cl:creator": "%s[service-name='%s']" % (service_name, instance_name),
        "nc:operation": "replace",
        "xmlns:cl": "http://clicon.org/lib",
    }

    for instance in root.services.ssh_users:
        if str(instance.service_name) != instance_name:
            continue

        devices = set()
        if "devices" in instance:
            service_devices = instance.devices
            for device in service_devices:
                devices.add(str(device))

        if "device_groups" in instance:
            for group_name in instance.device_groups:
                group_devices = get_devices_from_group(root, group_name)
                for device in group_devices:
                    devices.add(str(device))

        for user in instance.user:
            username = get_value(user, "name", required=True)
            user_class = get_value(user, "class", default="read-only")
            full_name = get_value(user, "full-name", default=False)
            ssh_key = get_value(user, "ssh-key", required=True)

            for device in root.devices.device:
                if devices:
                    if device.name.get_data() not in devices:
                        continue

                new_user = Element("user", attributes=attributes)
                new_user.create("name", data=username)
                new_user.create("class", data=user_class)
```

SSH users service

```
snc@snc-lab: ~
snc@snc-lab> show connections
Name                    State      Time                    Logmsg
=============================================================================================================
ptx-ac-1                OPEN       2024-09-09T09:17:02
ptx-ac-2                OPEN       2024-09-09T09:17:02
ptx-ac-3                OPEN       2024-09-09T09:15:16
ptx-ac-4                OPEN       2024-09-09T09:15:14
ptx-ac-5                OPEN       2024-09-09T09:15:14
snc@snc-lab> configure
snc@snc-lab[/]# set services ssh-users markus-test user markus
snc@snc-lab[/]# set services ssh-users markus-test user markus ssh-key "ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AA
snc@snc-lab[/]# set services ssh-users markus-test user markus full-name "Markus Krogh"
snc@snc-lab[/]# show compare
   <services xmlns="http://clicon.org/controller">
+     <ssh-users xmlns="http://clicon.org/ssh-users">
+        <service-name>markus-test</service-name>
+        <user>
+           <name>markus</name>
+           <full-name>Markus Krogh</full-name>
+           <ssh-key>ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIIOEg7Z4LTBUjAbI+U6Ct12Gee8+I2VWI0V4/E64GnM8 test
+        </user>
+     </ssh-users>
   </services>
snc@snc-lab[/]#
```

File: ssh_users.py

```python
from clixon.element import Element
from clixon.helpers import get_devices_from_group, get_path, get_value

SERVICE = "ssh-users"


def setup(root, log, **kwargs):
    try:
        _ = root.services.ssh_users
    except Exception:
        return

    service_name = SERVICE
    instance_name = kwargs["instance"]

    attributes = {
        "cl:creator": "%s[service-name='%s']" % (service_name, instance_name),
        "nc:operation": "replace",
        "xmlns:cl": "http://clicon.org/lib",
    }

    for instance in root.services.ssh_users:
        if str(instance.service_name) != instance_name:
            continue

        devices = set()
        if "devices" in instance:
            service_devices = instance.devices
            for device in service_devices:
                devices.add(str(device))

        if "device_groups" in instance:
            for group_name in instance.device_groups:
                group_devices = get_devices_from_group(root, group_name)
                for device in group_devices:
                    devices.add(str(device))

        for user in instance.user:
            username = get_value(user, "name", required=True)
            user_class = get_value(user, "class", default="read-only")
            full_name = get_value(user, "full-name", default=False)
            ssh_key = get_value(user, "ssh-key", required=True)

            for device in root.devices.device:
                if devices:
                    if device.name.get_data() not in devices:
                        continue

                new_user = Element("user", attributes=attributes)
                new_user.create("name", data=username)
                new_user.create("class", data=user_class)
```

SSH users service

snc@snc-lab: ~

```
snc@snc-lab[/]# commit diff
ptx-ac-1:
         <login xmlns="http://yang.juniper.net/junos/conf/system">
+           <user>
+              <name>markus</name>
+              <full-name>Markus Krogh</full-name>
+              <class>read-only</class>
+              <authentication>
+                 <ssh-ed25519>
+                    <name>ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIIOEg7Z4LTBUjAbI+U6Ct12Ge
+                 </ssh-ed25519>
+              </authentication>
+           </user>
         </login>
ptx-ac-2:
         <login xmlns="http://yang.juniper.net/junos/conf/system">
+           <user>
+              <name>markus</name>
+              <full-name>Markus Krogh</full-name>
+              <class>read-only</class>
+              <authentication>
+                 <ssh-ed25519>
+                    <name>ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIIOEg7Z4LTBUjAbI+U6Ct12Ge
+                 </ssh-ed25519>
```

Templating

- Variable substitution
  - Early decision for no logic/jinja2

File: snmp_template.xml

```
<config>
  <devices xmlns="http://clicon.org/controller">
    <template nc:operation="replace">
    <name>deploy-snmp</name>
    <variables>
      <variable>
          <name>location</name>
      </variable>
    </variables>
        <config>
          <configuration xmlns="http://yang.juniper.net/junos/conf/root">
              <snmp xmlns="http://yang.juniper.net/junos/conf/snmp">
                <location>${location}</location>
                <contact>noc@sunet.se</contact>
                <community>
                    <name>public</name>
                    <authorization>read-only</authorization>
                </community>
              </snmp>
          </configuration>
```

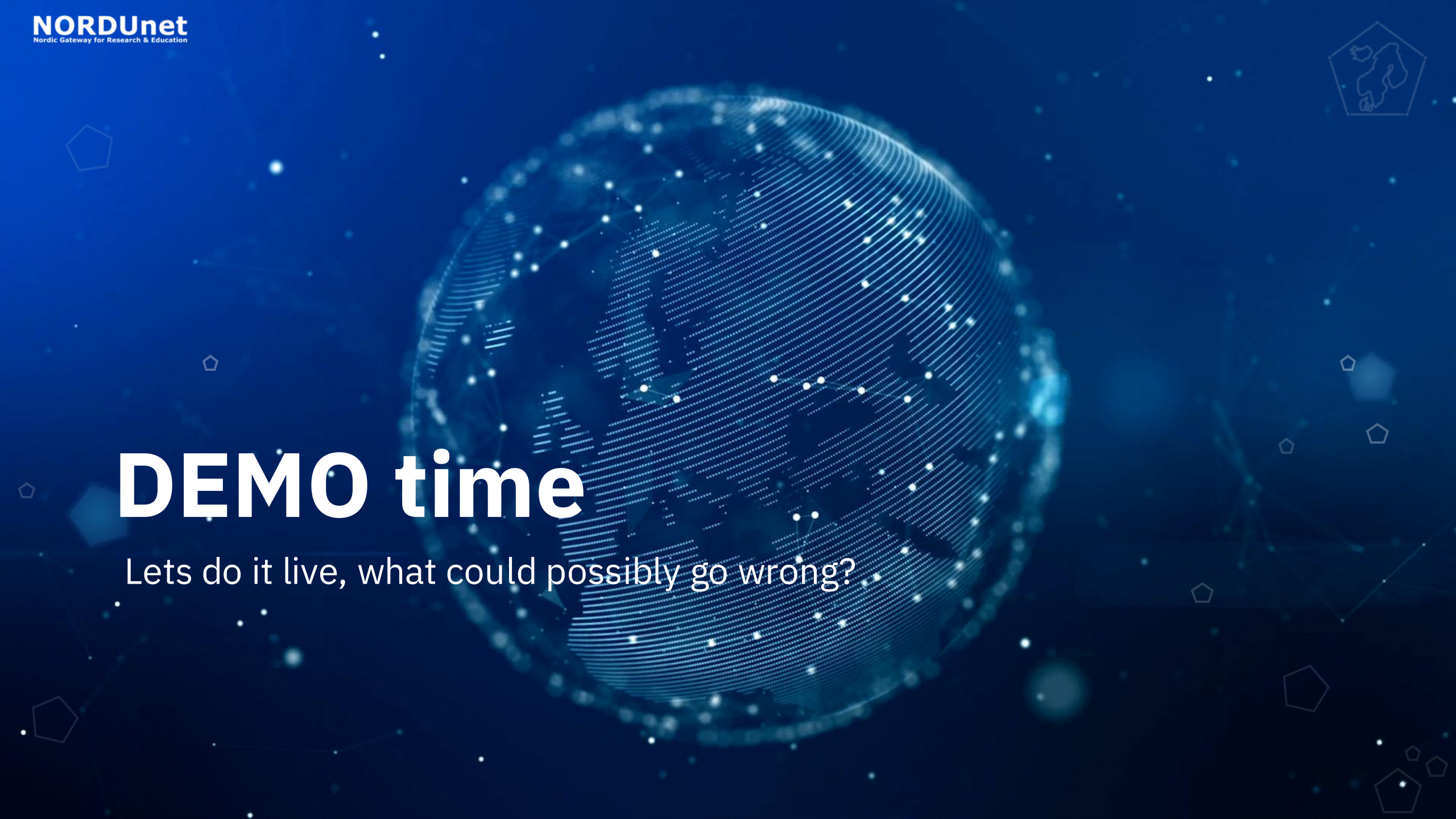
NORDUnet
Nordic Gateway for Research & Education

```
ces xmlns="http://clicon.org/controller">
emplate nc:operation="replace">
ame>deploy-snmp</name>
ariables>
 <variable>
     <name>location</name>
 </variable>
variables>
   <config>
     <configuration xmlns="http://yang.juniper.net/
         <snmp xmlns="http://yang.juniper.net/junos/
           <location>${location}</location>
           <contact>noc@sunet.se</contact>
           <community>
               <name>public</name>
               <authorization>read-only</authorizat
           </community>
         </snmp>
     </configuration>
   </config>
 </template>
ices>
>
```

Templating

```
snc@snc-lab[/]# apply template deploy-snmp ptx-ac-1 variables location Here
snc@snc-lab[/]# show com
 text                       xml                       |
snc@snc-lab[/]# show compare
                <snmp xmlns="http://yang.juniper.net/junos/conf/snmp">
-                 <location>location, city</location>
+                 <location>Here</location>
                </snmp>
snc@snc-lab[/]#
```

NORDUnet
Nordic Gateway for Research & Education



DEMO time

Lets do it live, what could possibly go wrong?

If demo gods not pleased

- Ascii cinema to the rescue

Lessons learned

- Got YANG?
  - Namespace clashes
  - Incomplete models
  - Large yang modles
- Reordering
  - Dynamic configuration created by services must be inserted in the correct order (according to device) to avoid unnecessary reordering and large diffs
- Scaling

Future work

- Tutorial and easier deployment
- More services
- Optimisations
- Integration with other systems
  - **Network Inventory**
  - **Post commit hooks**
  - **Workflow Orchestrator?**

Come play with us

- Join us at
  - https://matrix.to/#/#clixonforum:matrix.org
- Resources
  - https://github.com/clicon/clixon-controller
  - https://clixon-controller-docs.readthedocs.io
  - https://sunet.se/snc