Quantum Key Distribution Training CerberisXG

Version 06-September 2022
For Q2-2022 (V3.2.0 version)

Company presentation

SWISS QUANTUM

Founded in 2001

Geneva, Switzerland
Seoul, South Korea
Boston, USA
Vienna, Austria

By 4 quantum physicists from the University of Geneva

~100 employees in CH, including 50 engineers/scientists

Investments in 2018 by SK Telecom & Deutsche Telekom

Develops technologies and products based on quantum physics within 2 business units:
- Quantum-Safe Security
- Quantum Sensing

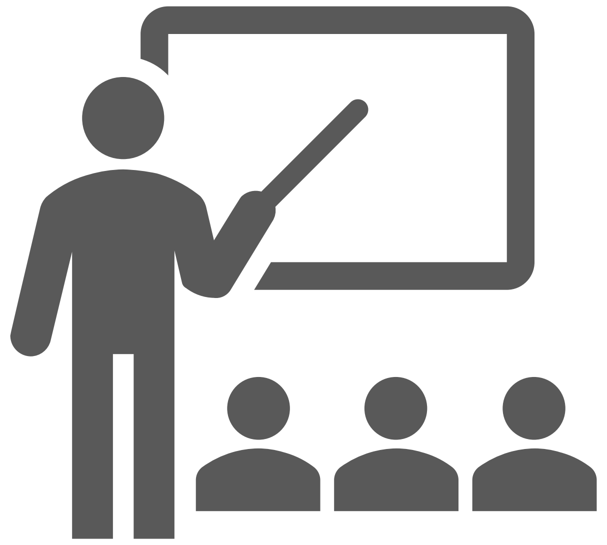
Performs R&D, production, professional services, integration, support

Clients: Governments / Banks / Gaming Industry / Universities / IT Security

Learning about QKD systems – What you must know

Goal of this section:

- Be able to understand the purpose of the QKD systems
- Be able to understand the different fiber connections used
- Be able to understand the purpose of the QMS server



CerberisXG presentation

- Commercial name: **CerberisXG / CerberisXGR**
- Type: **QKD** (Quantum Key Distribution)- 4th Generation
- Made in **Switzerland**
- Interoperability with major Ethernet and OTN encryptors
- Centrally monitored solution
- True (Quantum) random key generation
- Resilient to mechanical vibrations and thermal changes in fiber optics (polarization-independent scheme)

CerberisXG presentation

SWISS QUANTUM

Key Markets

- Telecom and Data Center Service Providers
- Financial Services Companies
- Governments and Defence
- Healthcare Organizations
- Critical Infrastructure
- IP-rich Enterprises

Key Applications

- Data center interconnections
- Metropolitan backbone optical networks
- Long distance distribution using relay nodes
- Key distribution across a complex network (ring, hub and spoke, meshed)
- Crypto keys as-a-service
- Validation of QKD and encryption pilot networks

IDQ

CerberisXG presentation

SWISS QUANTUM

Cerberis XG QKD System at a glance

Model	Cerberis XG
KEY FEATURES	
Key generation rate	1.25GHz pulse repetition rate
High speed hardware-based key processing, to distill the secret keys	✓
Key security parameter[1]	$\varepsilon_{QKD} = 4.\ 10^{-9}$
Dynamic range	12 dB (up to 16/18 dB on request)
Maximum length of quantum channel (typ. @ 0.23dB/km)	50 km (up to 70/80 km on request)
Secret key rate	2kb/s (12dB)
PHYSICAL PARAMETERS	
Dimensions	19" rackmount chassis; 22.4"
Dimensions (without front & back handles, and mounting kit)	W 428mm x L 610mm x H 43.6mm
Interfaces	Full Status LEDs, 2x Duplex Fiber SFP (Service Channel, KMS-O), 1x Simplex Fiber (Quantum Channel), 4x 1Gb Ethernet ports (Keys / Encryptors, KMS, Management, Aux), 1x RS-232 Serie (Console), 1x USB 2.0
Power supply	1+1 Redundant power supply consisting of two 300W hot-swappable power modules with input ratings of 100V-240Vac, 47-63Hz, 5-2.5A
Weight for one node	13.5 kg
Operating conditions	
Temperature	10 to 35°C
Max relative humidity (@35°C)	80% (non condensing)
Non-operating conditions	
Temperature	-10 to +60°C
Max relative humidity (@40°C)	90% (non condensing)
MANAGEMENT AND MONITORING	
Alerting functions & continuous monitoring	Cerberis XG can be administrated, configured and monitored via multiples interfaces (QNET REST Web API, QNET CLI Tools, QMS Web Application) allowing users or systems to control, and automatize through scripting, the various QKD services.

[1] The key security parameter characterises quantitatively the quality of the distributed keys. Technically, it is defined as the probability that the key distillation process went wrong, with either an error or at least one bit of the key leaked to the eavesdropper. It is normally calculated over a large block size, to allow an efficient distillation process. With the above value, the probability that an eavesdropper knows at least one bit of a 256-bits AES key is about 10^{-12}.
See for example: https://doi.org/10.1088/1367-2630/16/1/013047

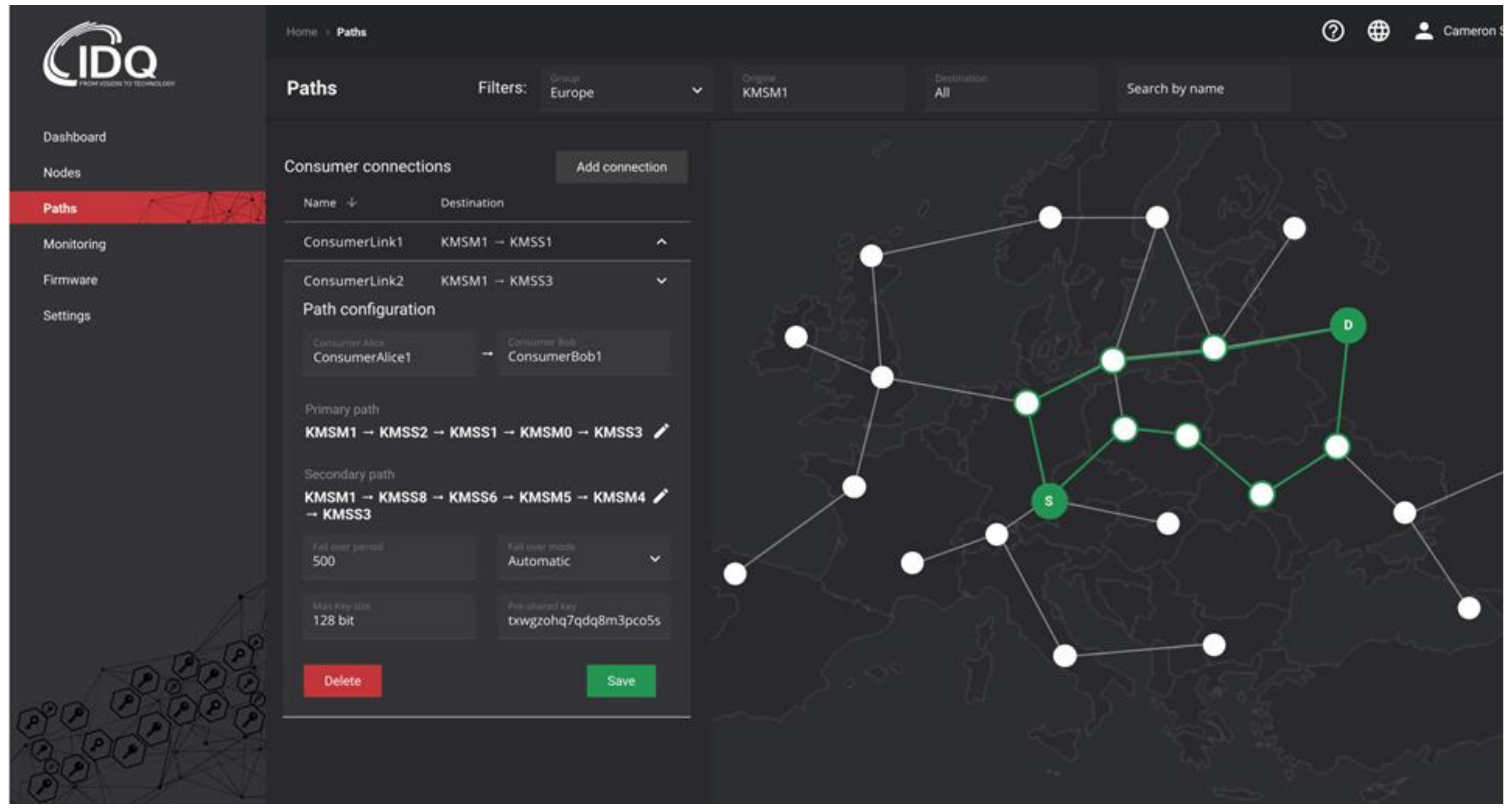
[2] For a typical installation with the ATCA chassis, the QNC and one QKD blade

At a Glance

SWISS QUANTUM

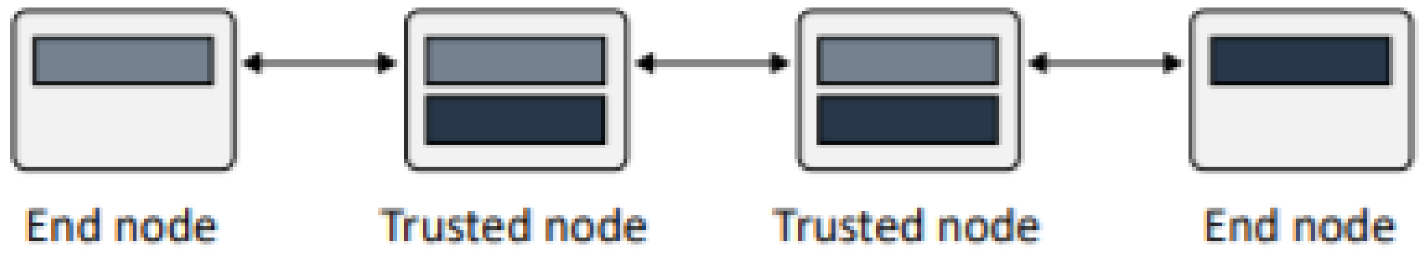
- **4th product generation**
- **<u>1U chassis</u>** (Alice or Bob)
- **High performance SPD**
- Quantum channel: C-band or O-band
  (WDM without additional filter)
- High availability (Dual PSU, Hot swap battery & fans module)
- **Separated key and management interfaces**
- **Trusted security** (Tamper detection, Secure memory module, IDQ20MC1 QRNG chip)

- Embedded KMS and management tools
- Enhanced monitoring (Syslog, SNMP, QMS …)
- New centralized EMS User Interface – **QMS**

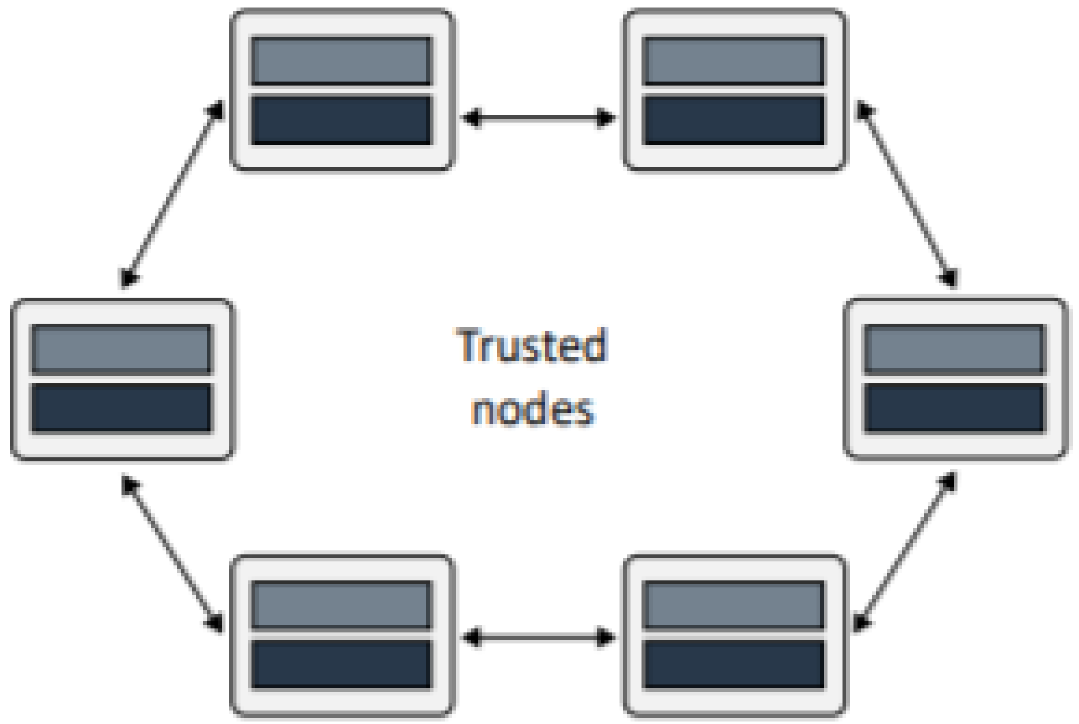


Supported Topologies

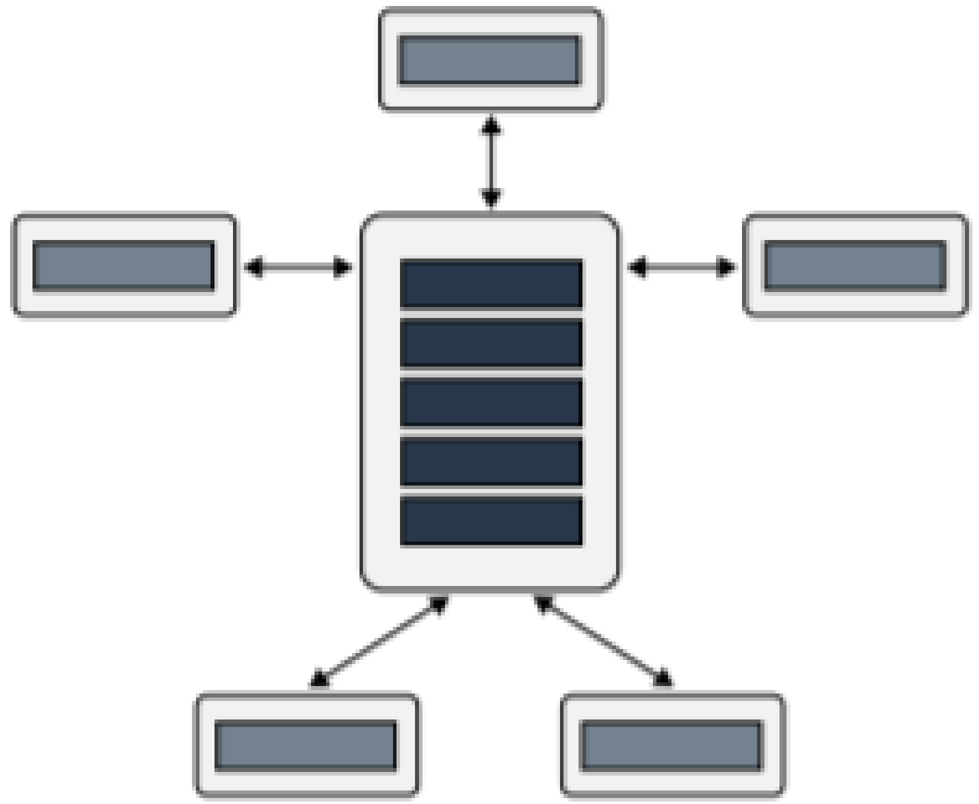
Point-to-point (with relay for long distance)

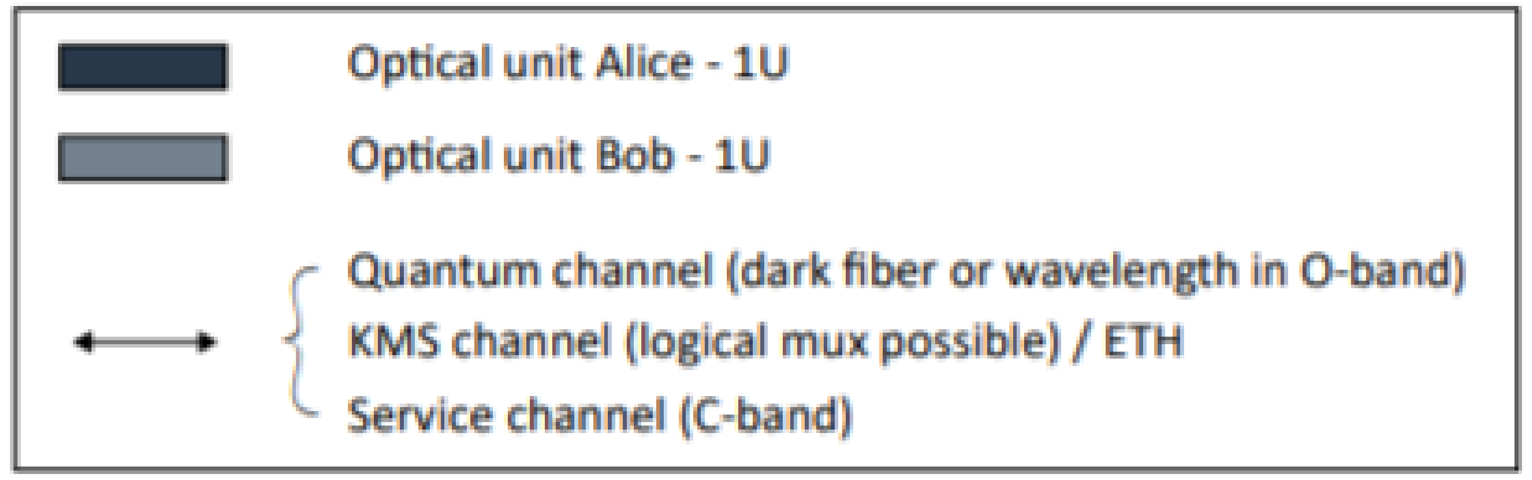


Ring network



Star





QKD - Basics

- Used **to exchange quantum bits** through a Quantum Channel
- Due to the nature of the Quantum Physics, **any attempt of hacking is detected**
- Quantum Channel **must use dark fiber – See IDQ Rep for alternate DWDM config**
- Quantum Channel **cannot work** through **actives devices**
- Quantum Channel cannot be **used for long distances or worldwide networks without the use of a trusted node**
- QKD CerberisXG system **doesn't encrypt data**
- QKD CerberisXG can be interconnected with **third party encryptors**

QKD - Basics

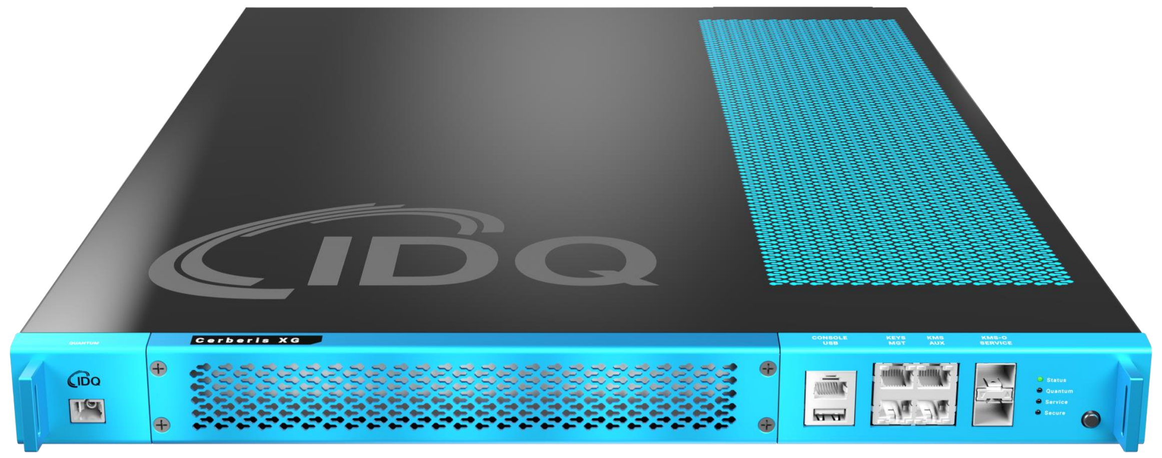
- The purpose of the QKD system is quite simple:

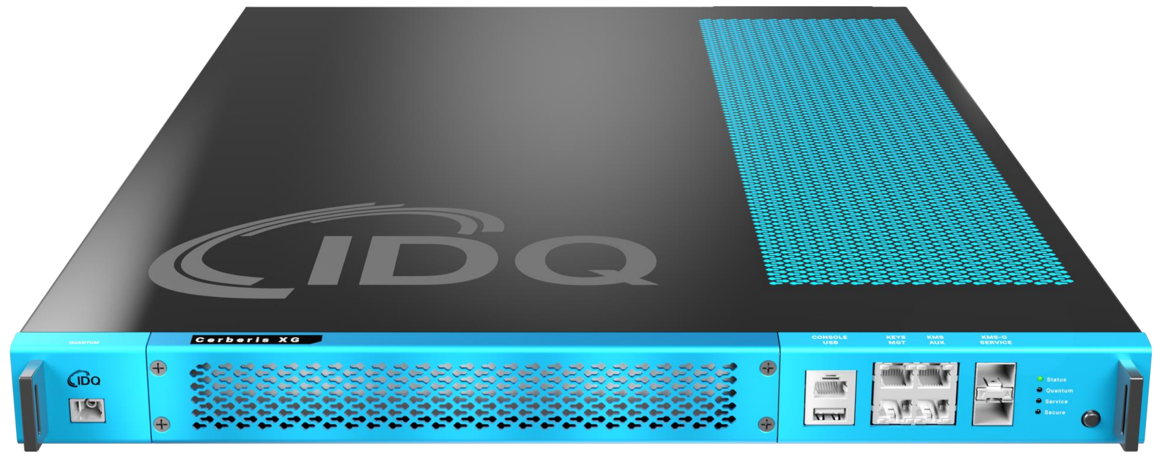
1. **Exchange quantum bits** on a quantum channel used to create keys
2. **Deliver keys** to third party devices (Encryptors or HSM)
3. Third party devices **use these keys to encrypt data**

QKD Topology

SWISS QUANTUM

Network Architecture



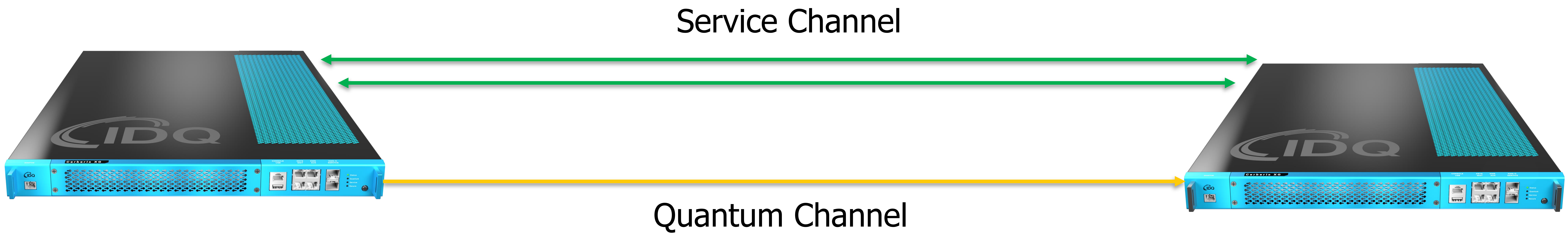


QKD Topology

- SMF (Single Mode Fiber)
- Uses quantum physics – Sends “Single Photons”
- Losses need to be minimized
- No active device between
- Unidirectional
- Support for C-Band and O-Band (1310nm). To be specified with order.

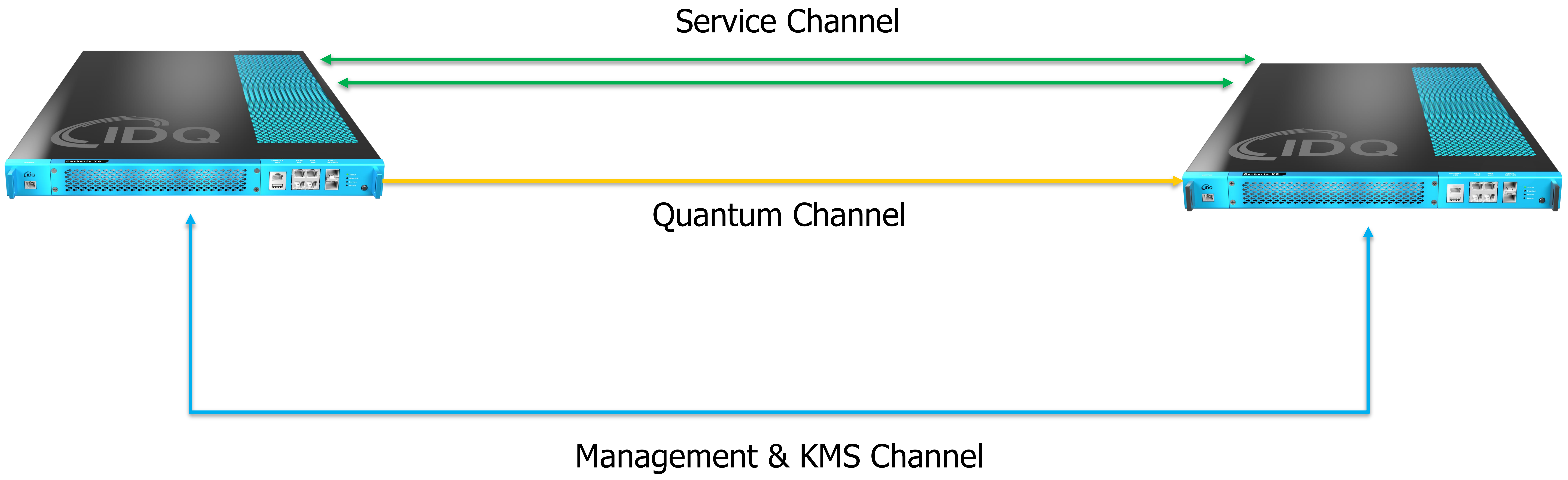
QKD Topology

- Dual SMF (Single Mode Fiber)
- 2.67 Gbps rate (Multirate Transceiver)
- IDQ-provided Finisar – spec'd to order wavelength based on architecture.
- Used for QKD clock synchronization and post-processing
- Bidirectional
- Not Ethernet – Proprietary Protocol. Cannot pass through switches/routers



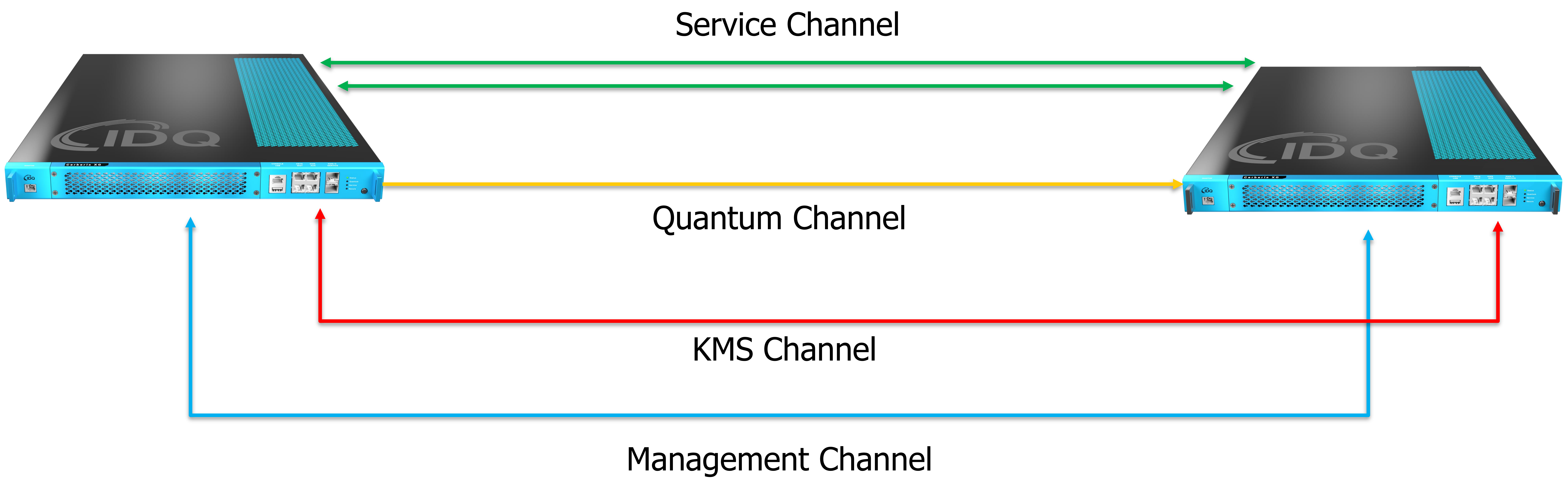
QKD Topology

- Ethernet RJ45
- Can use existing management network or dedicated network
- KMS traffic cannot use NAT (Network Address Translation)
- Management traffic can use NAT



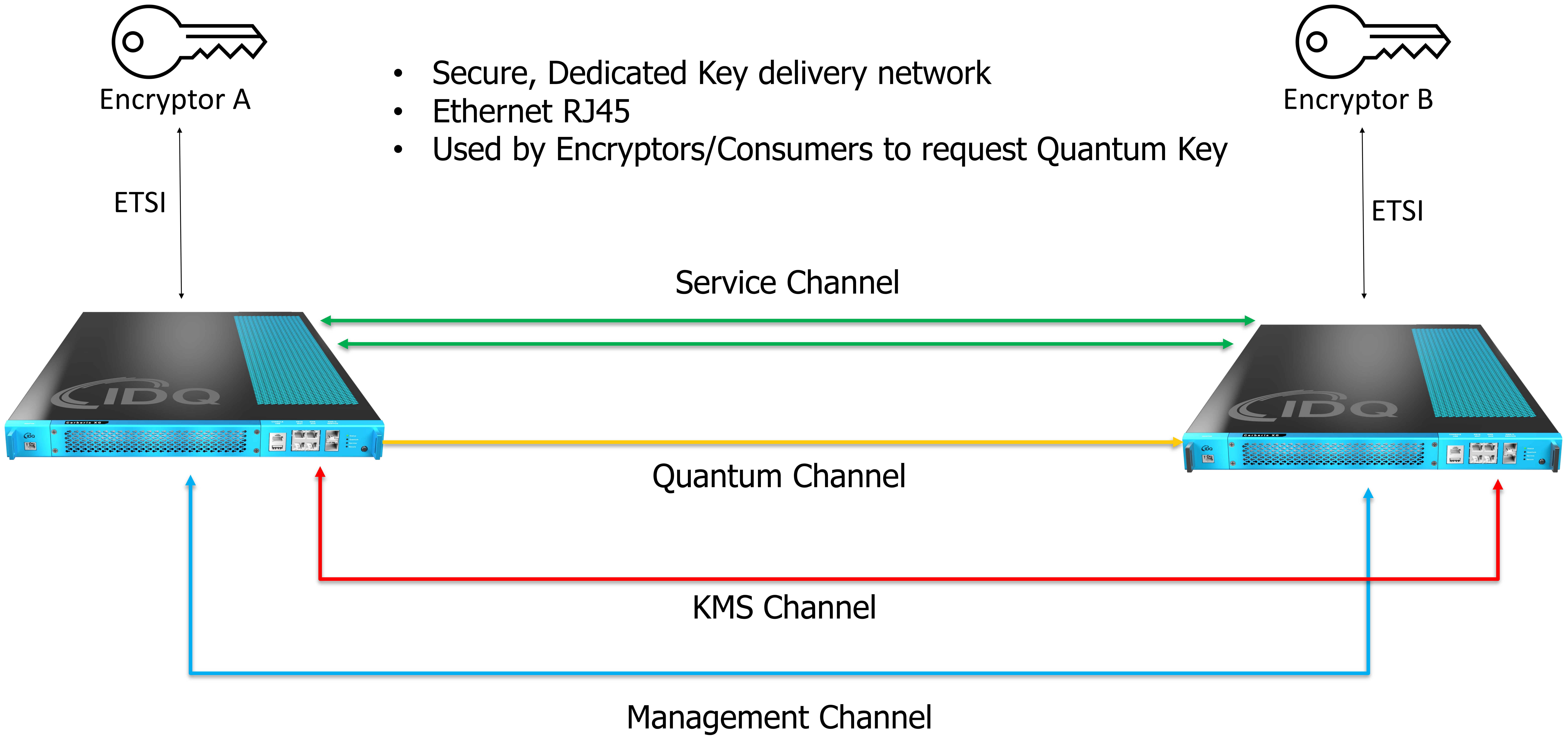
QKD Topology

- Optionally separate KMS traffic to dedicated network for added security
- RJ45 or Optical SFP

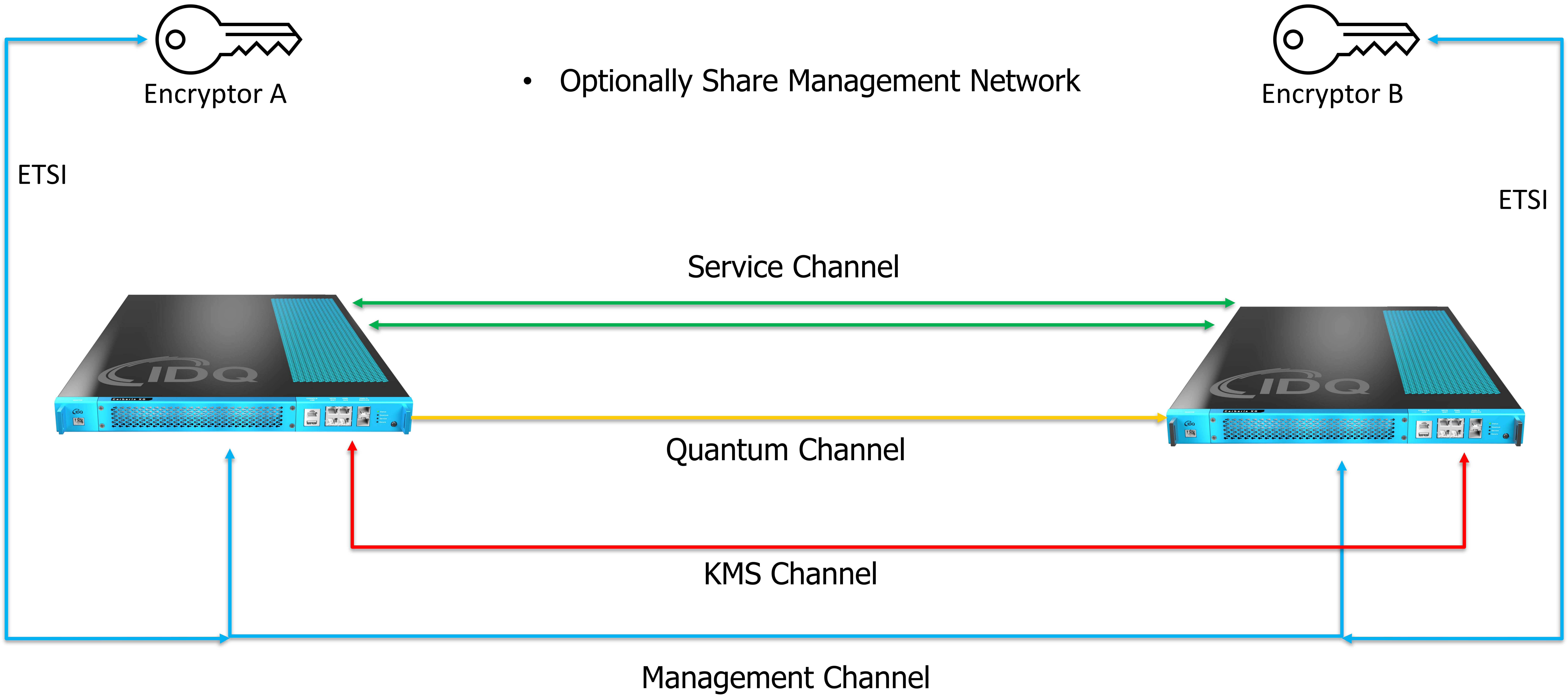


QKD Topology

- Secure, Dedicated Key delivery network
- Ethernet RJ45
- Used by Encryptors/Consumers to request Quantum Key



QKD Topology



QKD Topology

It is Possible to Optically Mux Traffic Together

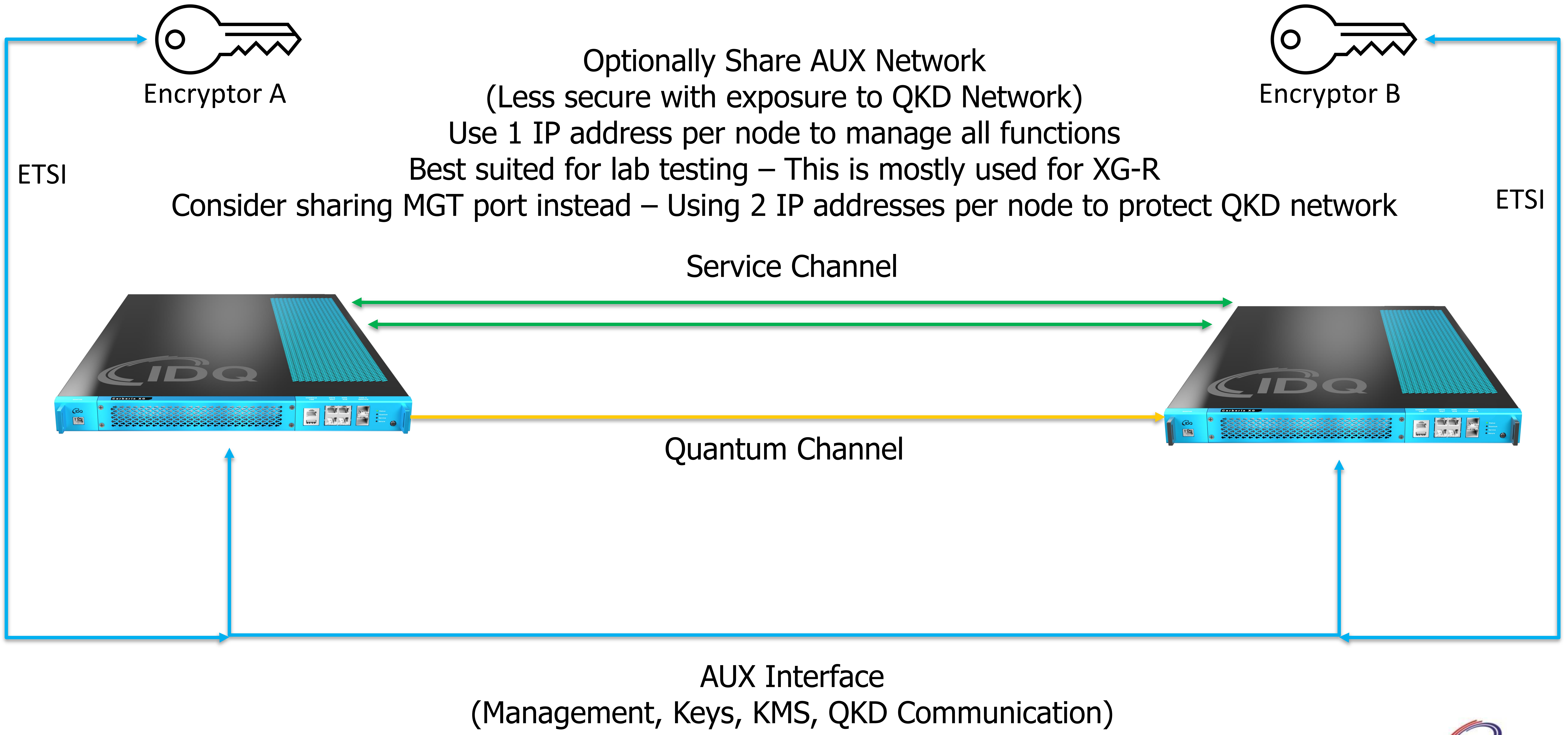
<u>For Multiplexing Quantum Channel in the O-Band, Consult IDQ</u>

QKD Topology

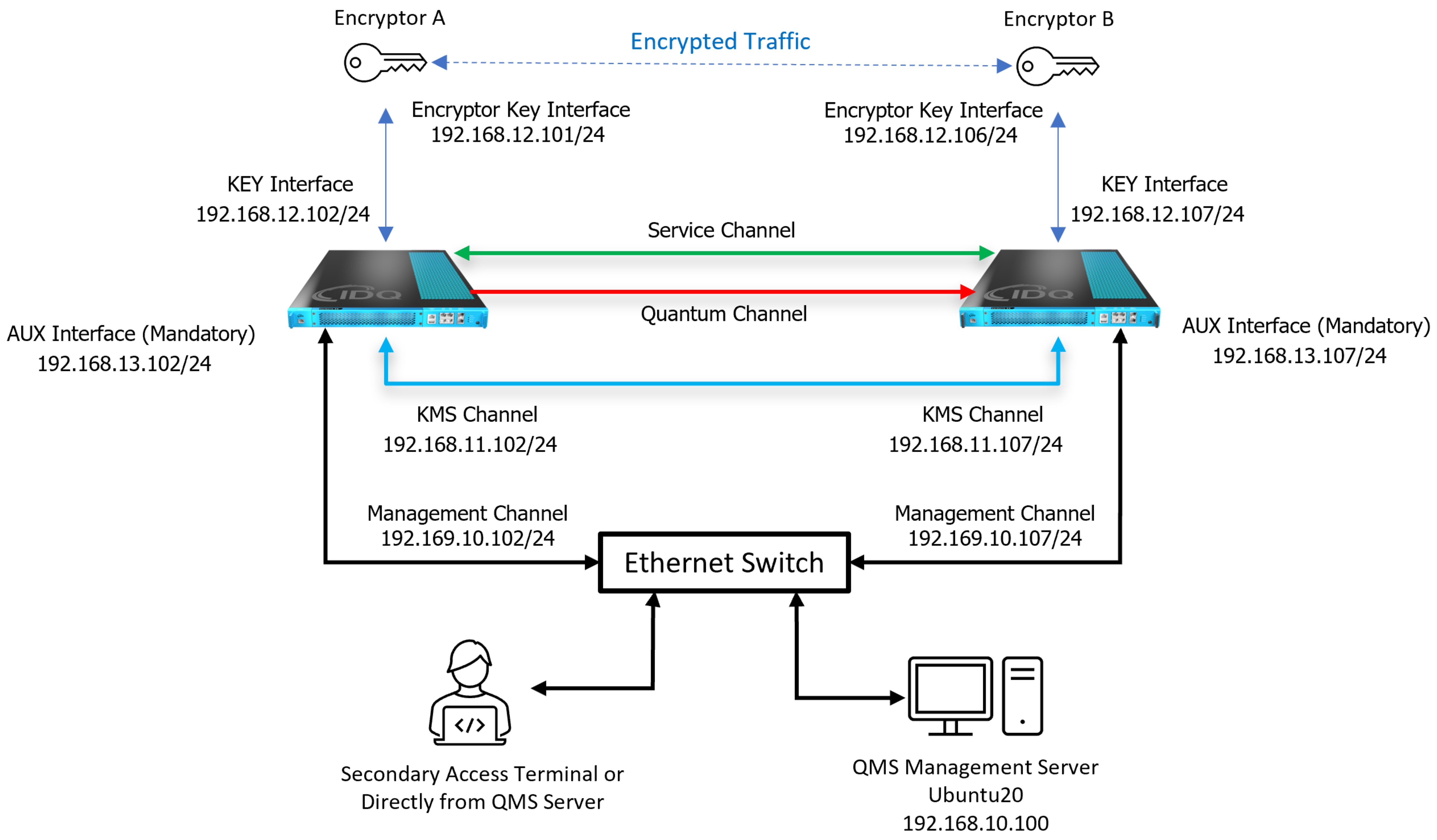
It is Possible to Optically Mux Traffic Together

Including the Encrypted Data Traffic

QKD Topology

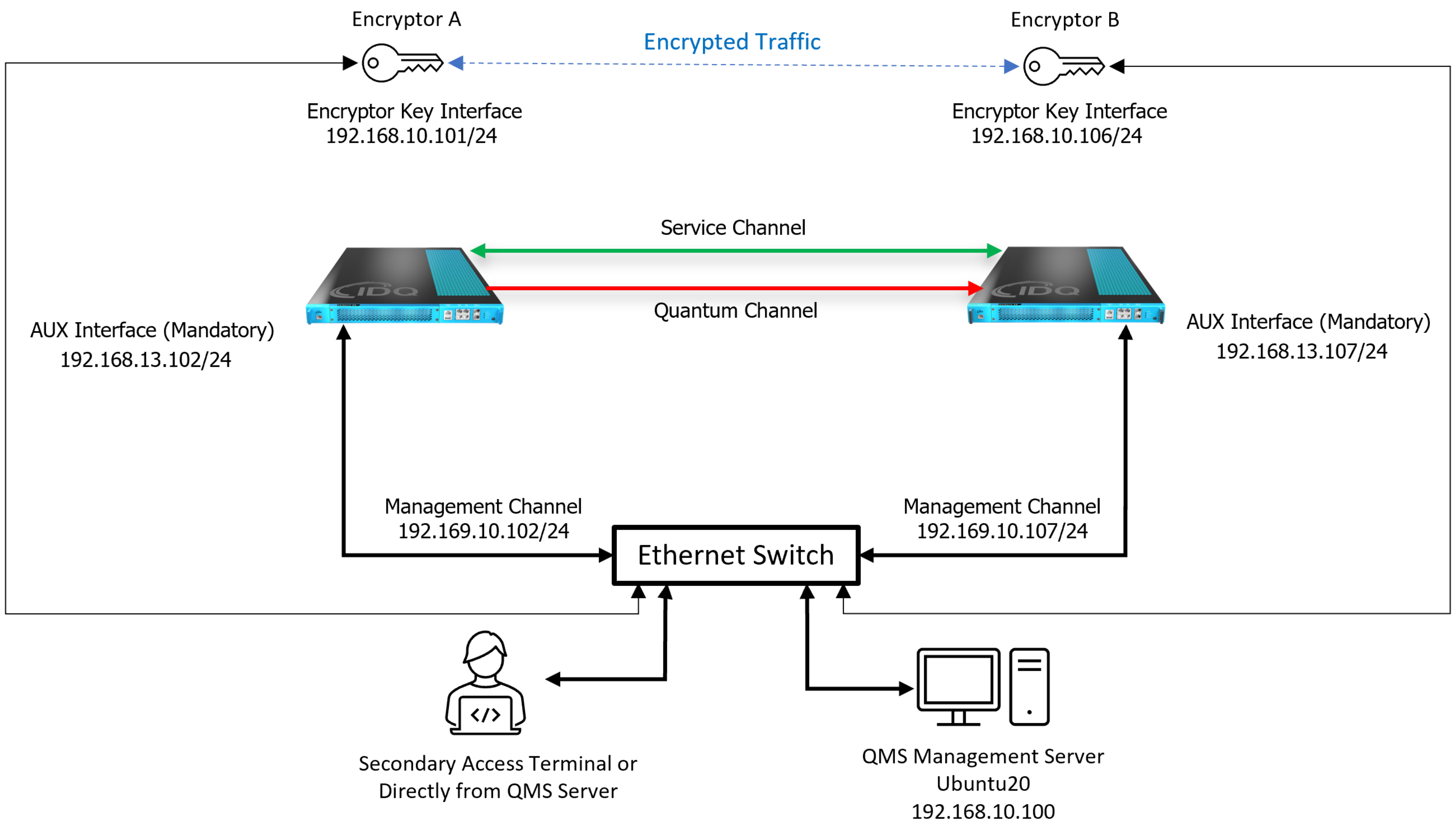


Separate Networks for Management, Key, KMS, AUX



MGT Alice:	192.168.10.102/24
KMS Alice:	192.168.11.102/24
KEY Alice:	192.168.12.102/24
AUX Alice:	192.168.13.102/24
Encryptor Alice:	192.168.12.101/24
MGT Bob:	192.168.10.107/24
KMS Bob:	192.168.11.107/24
KEY Bob:	192.168.12.107/24
AUX Bob:	192.168.13.107/24
Encryptor Bob:	192.168.12.106/24
QMS Server:	192.168.10.100/24

Shared Networks for Management, Key, KMS. (Internal AUX Mandatory.)



MGT Alice: 192.168.10.102/24
AUX Alice: 192.168.13.102/24

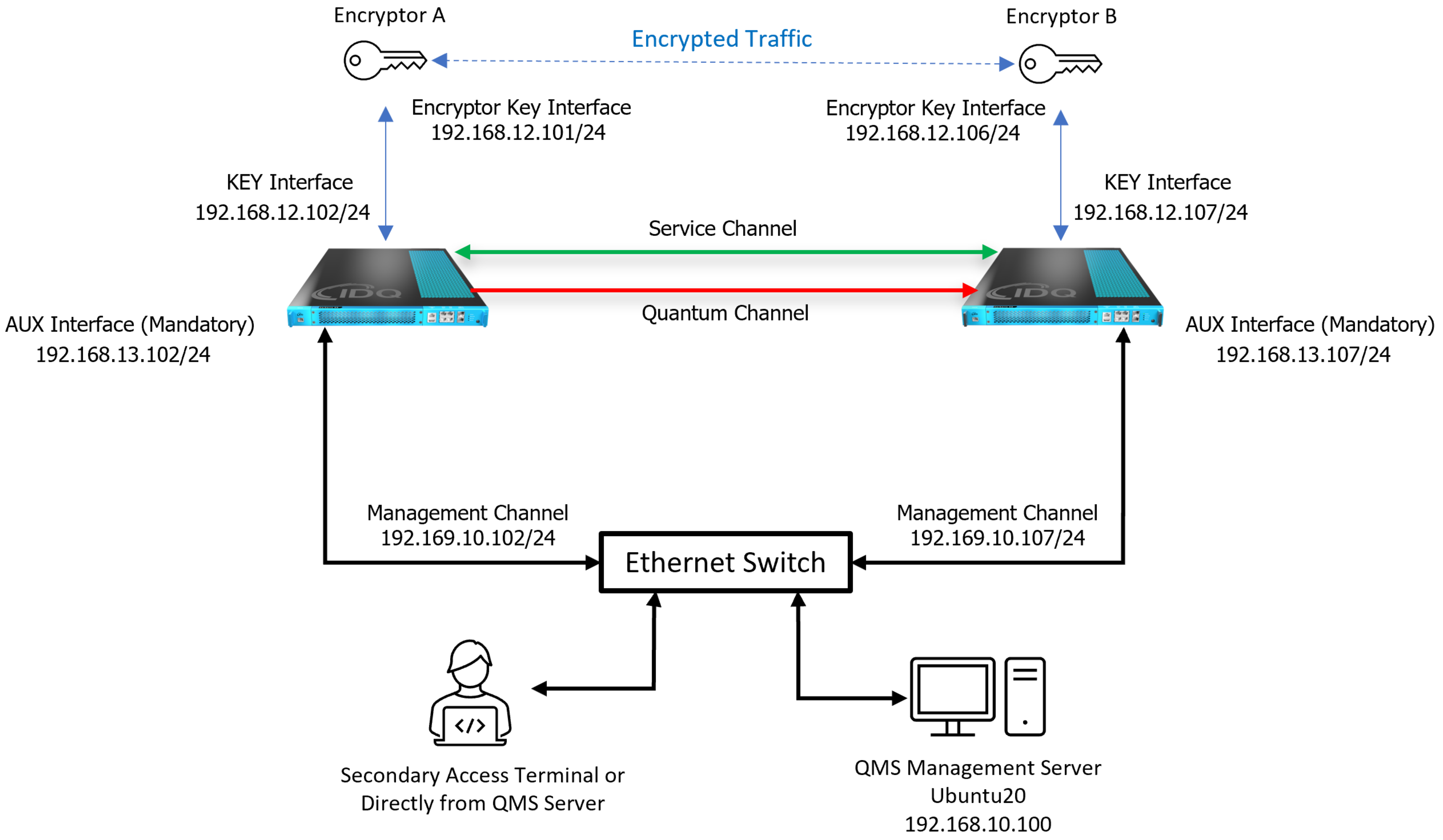
Encryptor Alice: 192.168.10.101/24

MGT Bob: 192.168.10.107/24
AUX Bob: 192.168.13.107/24

Encryptor Bob: 192.168.10.106/24

QMS Server: 192.168.10.100/24

Shared Networks for Management, KMS. Separate Network for KEYS (Internal AUX Mandatory.)



MGT Alice: 192.168.10.102/24
KEY Alice: 192.168.12.102/24
AUX Alice: 192.168.13.102/24

Encryptor Alice: 192.168.12.101/24

MGT Bob: 192.168.10.107/24
KEY Bob: 192.168.12.107/24
AUX Bob: 192.168.13.107/24

Encryptor Bob: 192.168.12.106/24

QMS Server: 192.168.10.100/24

Default Route Limitation

At this time, only 1 default route is permitted across interfaces.
Last default route configured becomes active.

For example, you can configure interface #1, 2, 3, 4. The interfaces with no default route should have a default route configured of 0.0.0.0

Static routes are not supported.

If 2 interfaces need to communicate across different subnets (one for MGT and one for KMS), you could consider an L2 tunnel over L3 for the second interface as a workaround.

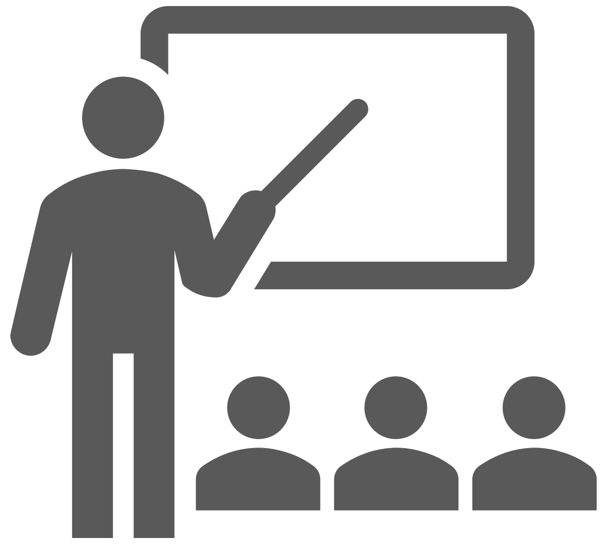
QKD – QMS/QNET

- QMS: **Q**uantum **M**anagement **S**ystem
- **Mandatory** to configure QKD system – QNET API also available
- Used to configure the QKDs, the paths and the encryptor integration
- Can be used on a **virtual machine** (VM) or physical server
- Low server performance requirement
- Simple deployment with Docker

QKD Chassis Details

Goal of this section:

- Be able to identify and understand each component of the system.



XG vs XGR

SWISS QUANTUM

CerberisXG - Commercial

- **QSafe & Resilient KMS**
  - Interface with majors encryptor vendors
  - Complex Network Topology support
- **QNet CLI (Embedded / Remote) interface**
  - Administration
  - Configuration
  - Monitoring
- **Web API for Configuration & Monitoring**
  - QKD
  - KMS
- **Support for Standard Monitoring protocols**
  - SNMP
  - Syslog
- **QMS Central Server & Web Application**

CerberisXGR - Academic

- **Same hardware platform as Cerberis XG (different color)**
- **Same KMS functionality as Cerberis XG**
- **Software differentiation**
  - **IDQ4P interface**
    - Key, Management and Signal
    - Access to raw key material
  - **Web API for Advanced Configuration & Monitoring**
    - Alice parameters (regulation, laser)
    - Bob parameters (data & monitor detectors)

Bob vs Alice

Which chassis is Bob???

Bob – **Grey** Handles

Alice – **Blue** Handles
Red for XGR

Front Panel

Alice & Bob – differentiation
Left and right handles
- Bob: grey
- Alice: blue (or red for XGR)

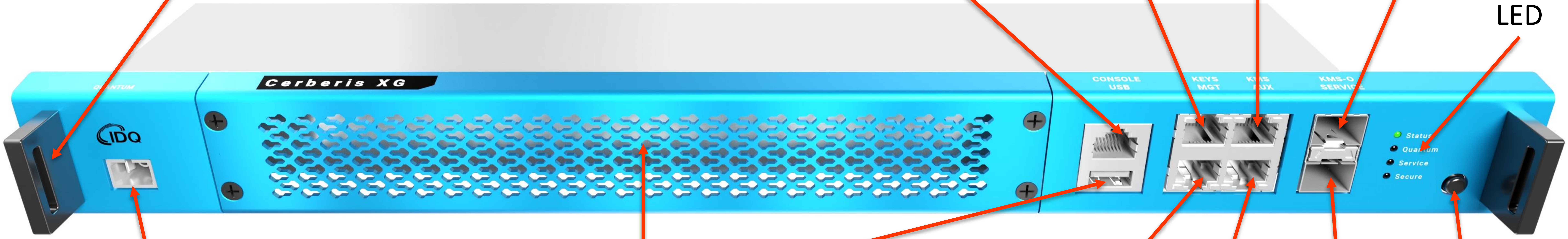
Serial
(RJ-45)

Key Delivery
(RJ-45)

KMS-O
(SFP)

KMS
(RJ-45)

Status
LED



USB
(For Upgrades and Log Copy)

AUX
(RJ-45)

Quantum Channel
(SC/UPC)

Hot-Swappable Fans & Battery

Management
(RJ-45)

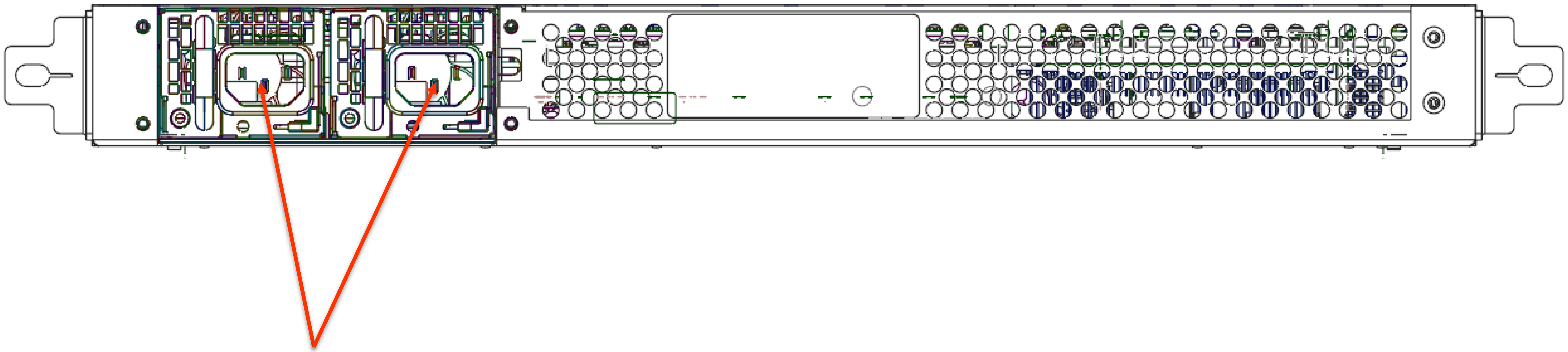
Power Switch

Service Channel
(SFP)

Status LEDs

- **Status**
  QKD status (e.g. Initialization, Alignment, Key Exchange)
- **Quantum**
  Quantum Channel status
- **Service**
  Service Channel status
- **Secure**
  Tamper detection switch status

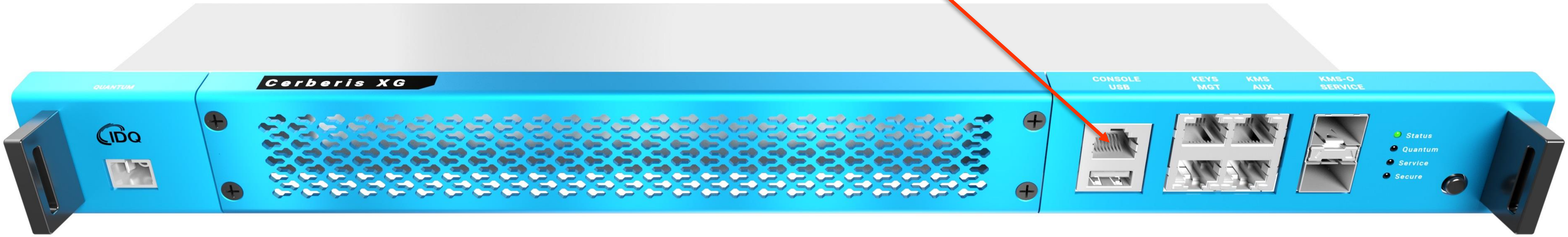
Rear



Dual Power Supplies

Serial Interface

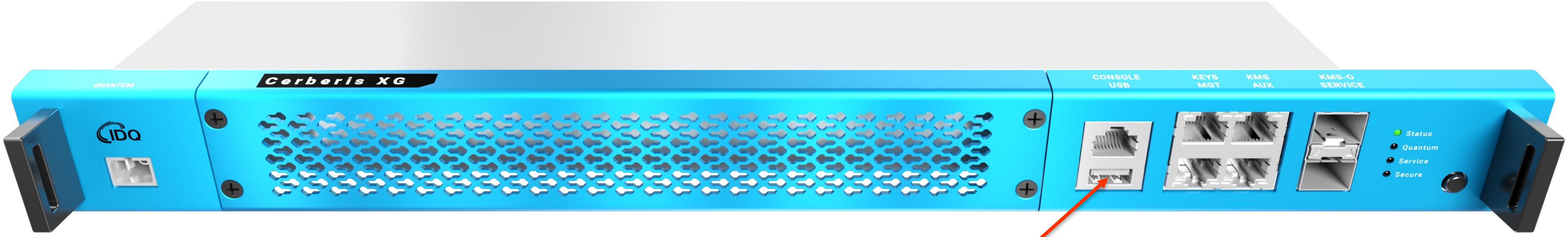
Serial
(RJ-45)



CLI via Serial interface used for initial configuration (IP addressing, etc)

- Speed bauds: 115200
- Date bits: 8 / Stop bits: 1 / Parity: none
- Flow control: XON/XOFF

Serial Interface

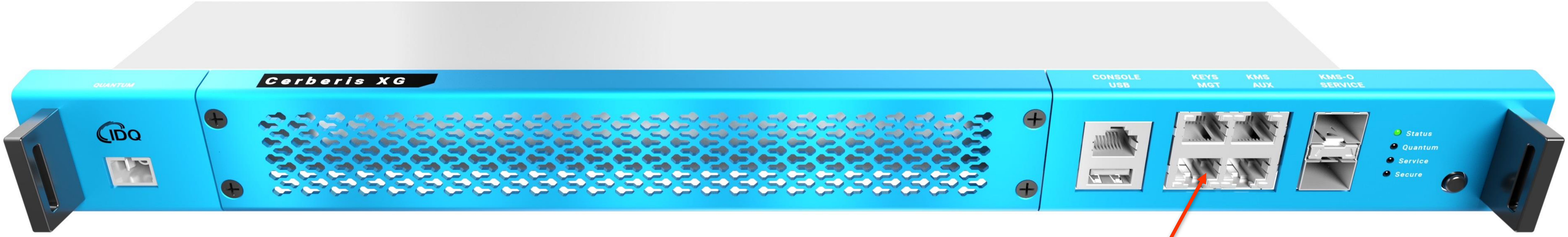


USB

USB Interface:

- System Upgrades
- Copy debug logs

MGT Interface

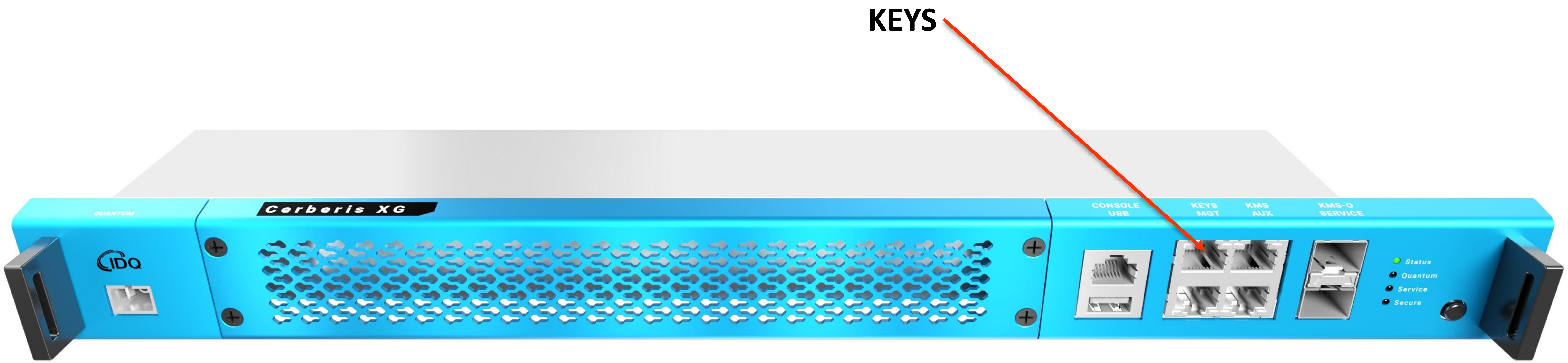


Management Interface

- SSH Access to CLI
- QMS/QNET Management Interface
- Optional KMS to KMS Communication
- SNMP Monitoring
- Syslog Forwarding
- Optional Key Delivery Interface

MGT
(RJ-45)

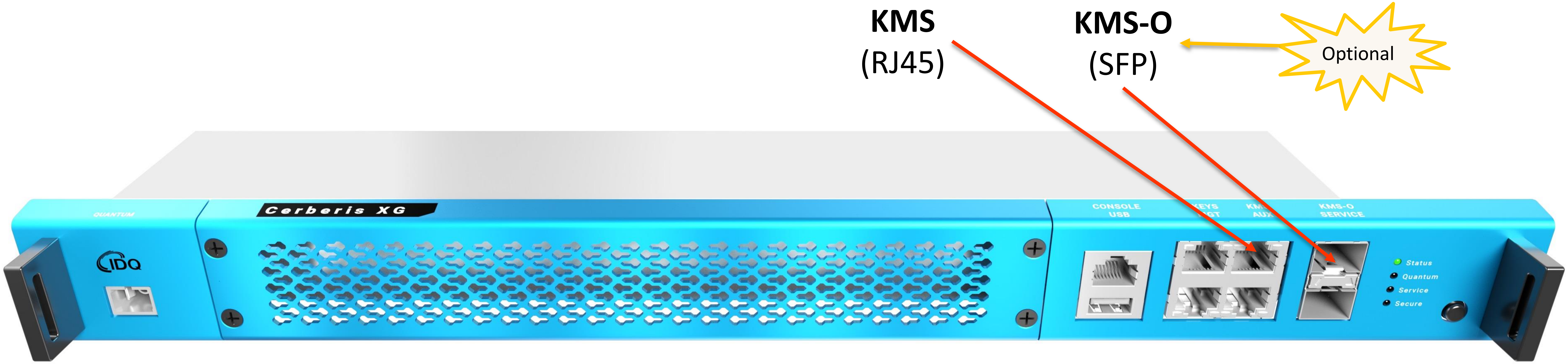
KEYS Interface



Dedicated KEYS Interface:

- Key Delivery Interface
  - Key Integration to 3rd Party Devices
  - ETSI 014 REST API, Etc.
  - Optionally use Shared MGT Interface Instead

KMS Interfaces

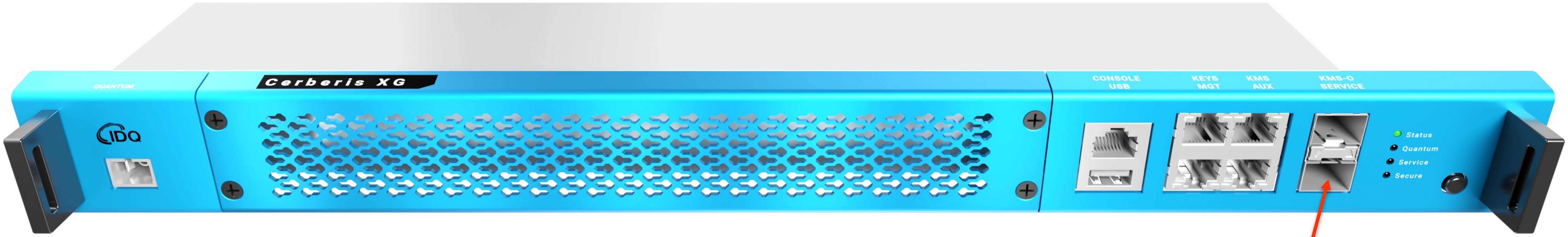


Dedicated KMS Interface (Key Management System):

- Chassis-to-Chassis KMS Communication
- Cannot Perform NAT (Must pass original IP address end-to-end)
- Default: Use RJ45 Port. Optional: Use Optical SFP (Must change interface type via CLI)
- Optionally use shared MGT Interface Instead

Service Channel

SWISS QUANTUM

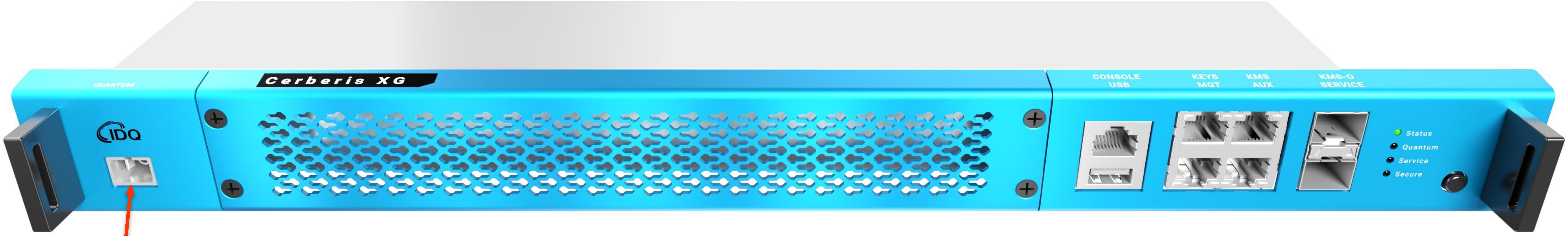


Service Channel:

- Not Ethernet, SONET, SDH – Proprietary
- Cannot Pass Through Switches or Routers
- Can be Muxed/DeMuxed with Other Traffic
- Used for Clock Synchronization & Data Post Processing
- Uses Multirate SFP
- Use SMF (Single Mode Fiber)
- Standard Config: 2x 1553.33nm (ITU Ch 30 C-band) and 80km or more
- Optional Config: 1x 1553.33nm (ITU Ch 30 C-band), 1x 1554.13nm (ITU Ch 29 C-band)

Service Channel
(SFP – LC/UPC)

Quantum Channel



Quantum Channel
(SC/UPC)

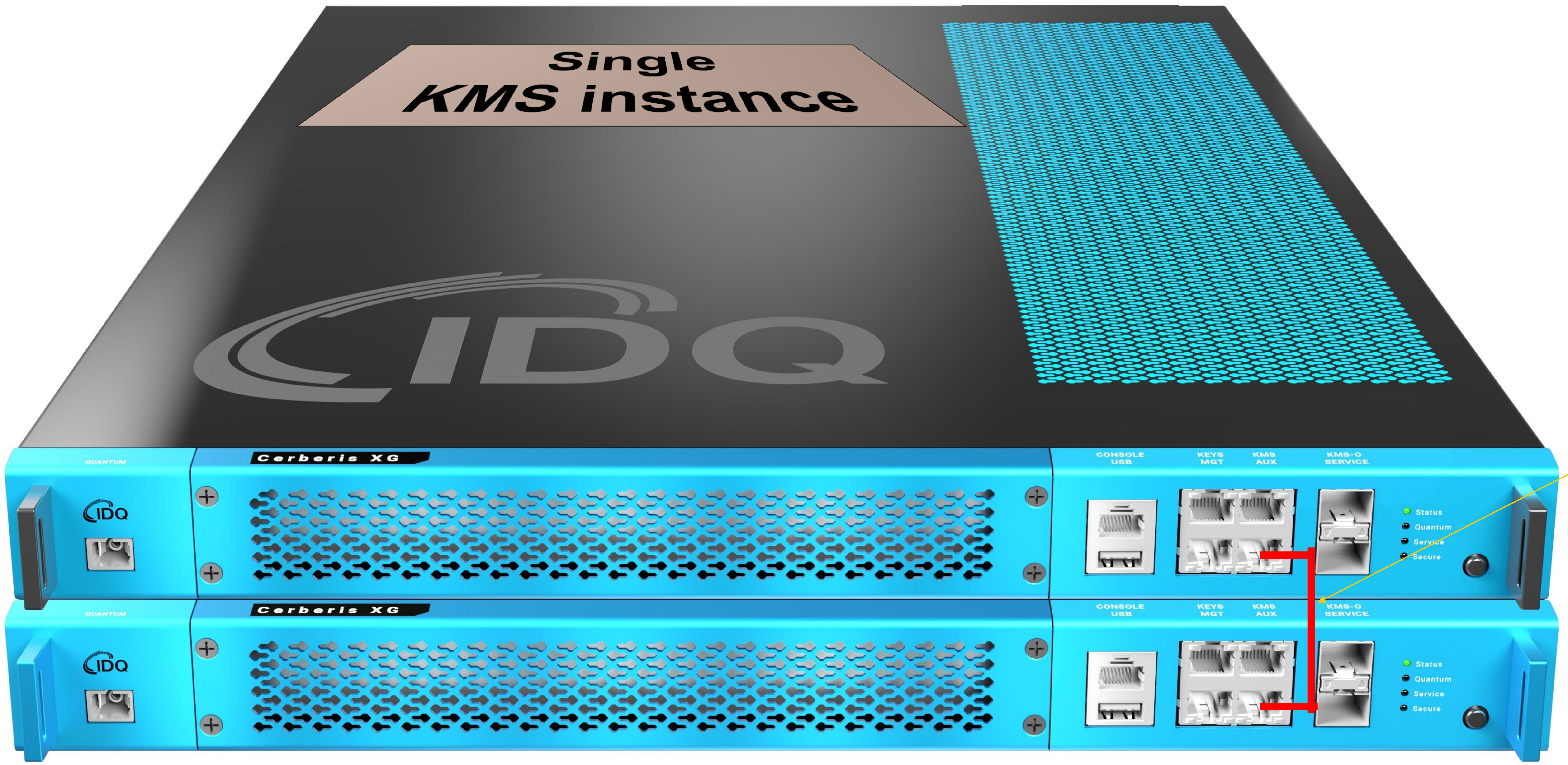
Quantum Channel:

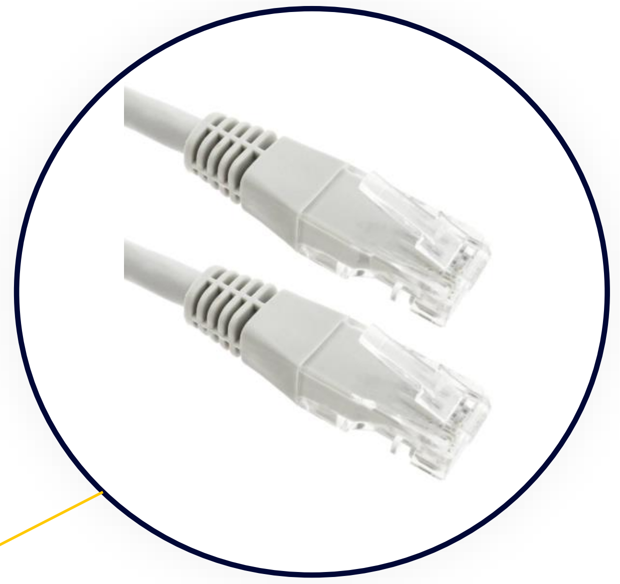
- Sends Single Photons – Uses Quantum Physics
- Default Wavelength 1551.72nm (ITU channel 32 of C-band)
- SC/UPC
- Use SMF (Single Mode Fiber)
- Must use Dedicated Dark Fiber – See IDQ Rep for Shared DWDM Fiber Configuration via 1310m
- Cannot Pass Through Active Devices Including Amplifiers, Switches or Routers
- Optional: Purchase DWDM-Capable System with Quantum Channel at 1310nm

AUX Interface – Trusted Node

Bob

Alice





Ethernet patch cord through “AUX” ports (IDQ4P protocol)

AUX Interface:

- When using a Trusted Node QKD Architecture, Connect the Alice & Bob QKD Systems Together Using the AUX Interface (RJ45)
- Use Only 1 Shared KMS Instance at This Node

NB: All Cerberis XG RJ-45 ports are auto MDI/MDIX

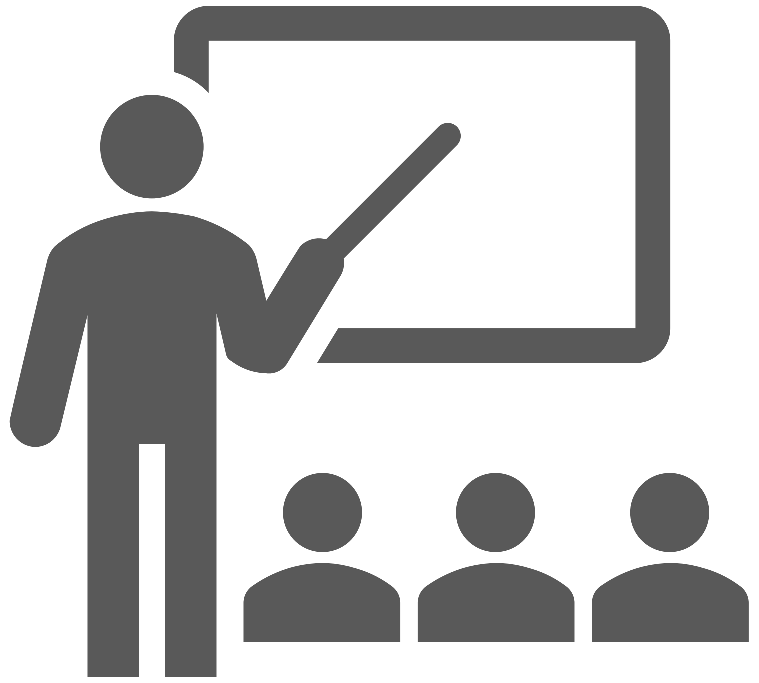
CERBERIS XG

Physical Installation step by step

Cerberis XG – Installation step by step

Goal of this part

- Be able to understand the safety requirements
- Be able to properly install a QKD system in a 19” rack
- Be able to make sure the system is properly started
- Be able to clean and connect the different fibers



Installation procedure – Safety precautions

SWISS QUANTUM

Safety precautions

- Carefully read through this procedure before operating the Cerberis XG system.
- This system is intended to only be used **indoors.**
- **Never look directly into an active optical fiber** and ensure proper eyewear is worn when handling unterminated fiber ends.
- Always handle the devices **using proper ESD** damage prevention and grounding methods.
- **Ensure all power connections are connected to a ground socket**. Failure to connect to ground creates a shock hazard which may cause injury to the operator.
- **Refrain from bringing water or other liquids around any part of the system**. If the systems do get wet, shut all devices down immediately.

IDQ

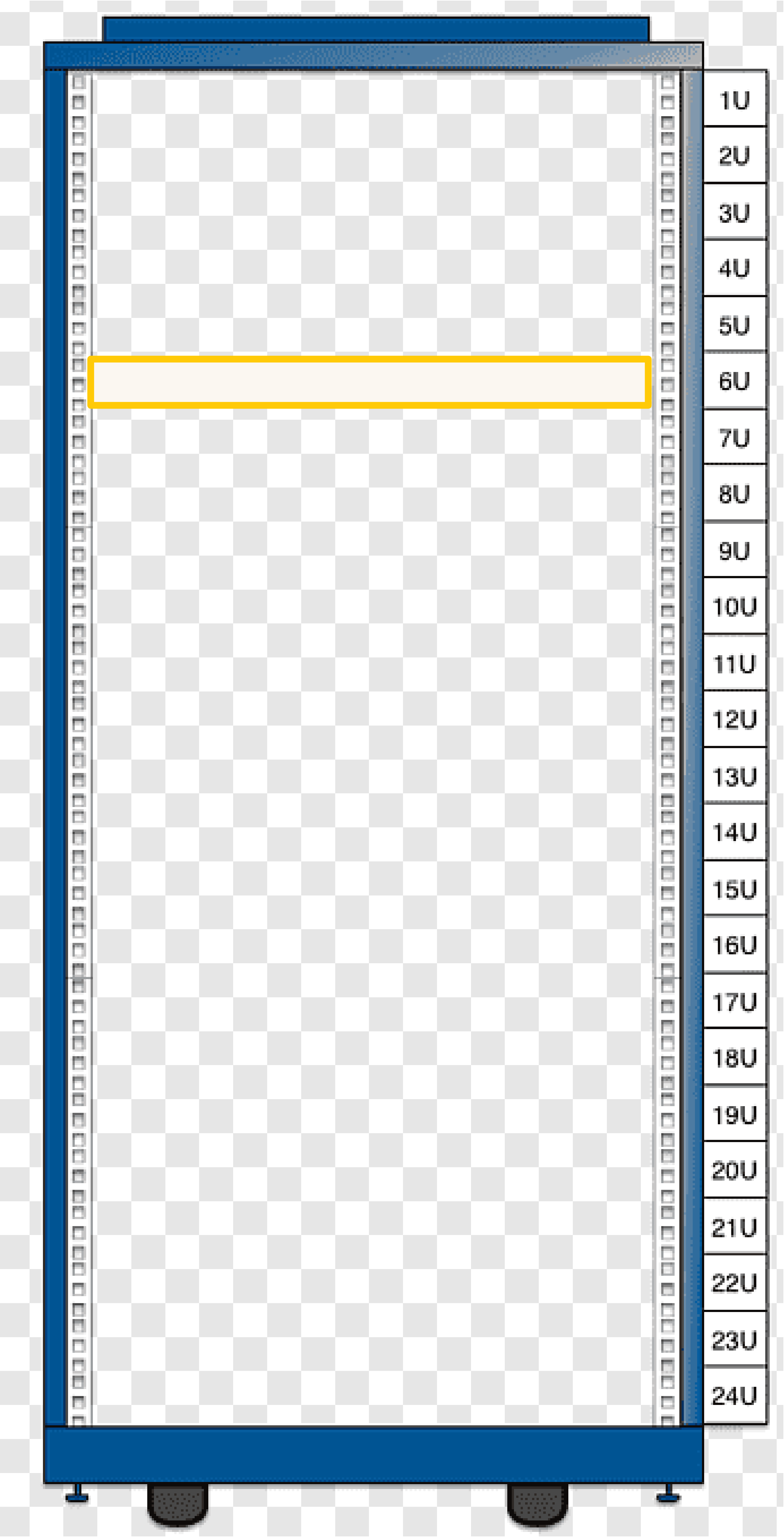
Delivery

SWISS QUANTUM

Verify the parcel is not damaged

IDQ

Cerberis XG – Physical setup (Rack mount)



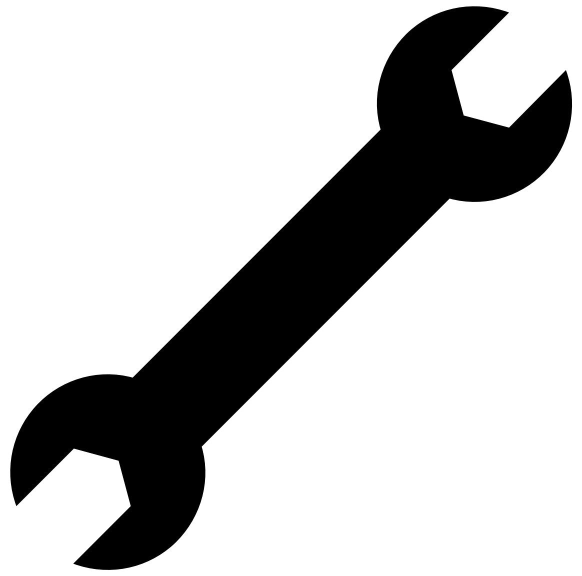
Only need to allocate 1U per device

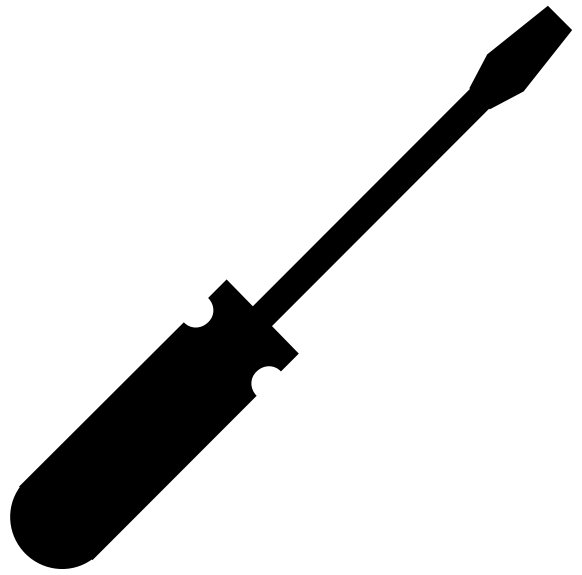
A. Fix the mounting bracket.
B. Fix the internal rail to the Cerberis XG/XGR
C. Fix the external rails to the rack.
D. Mount the Cerberis XG/XGR in the rack

Cerberis XG – Physical setup (Rack mount)

To install the Cerberis XG in a rack, the required tools are:

- Screwdriver Torx T10
- Screwdriver Torx T20
- Screwdriver Philips Head PH2
- 8mm wrench



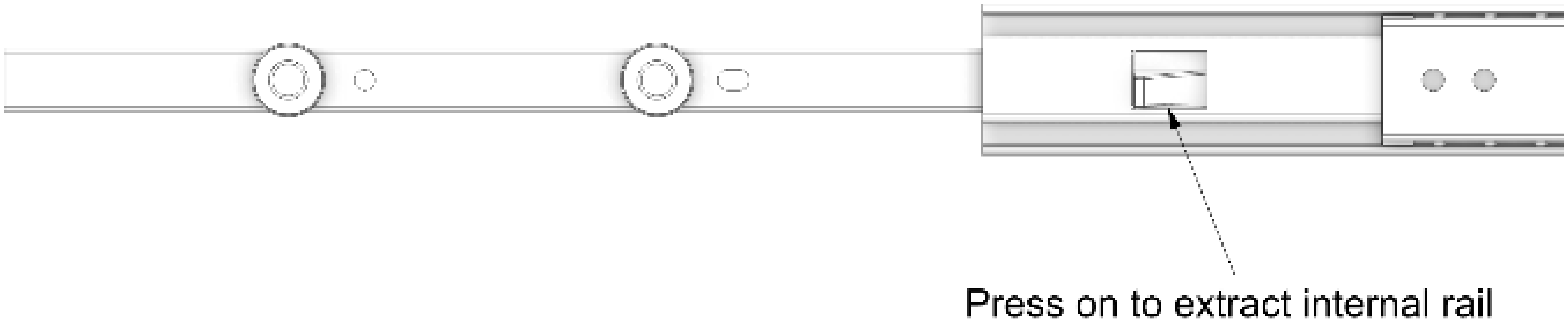


Cerberis XG – Physical setup (Rack mount)

SWISS QUANTUM

The installation of the rails can be done using the 6 steps described below:

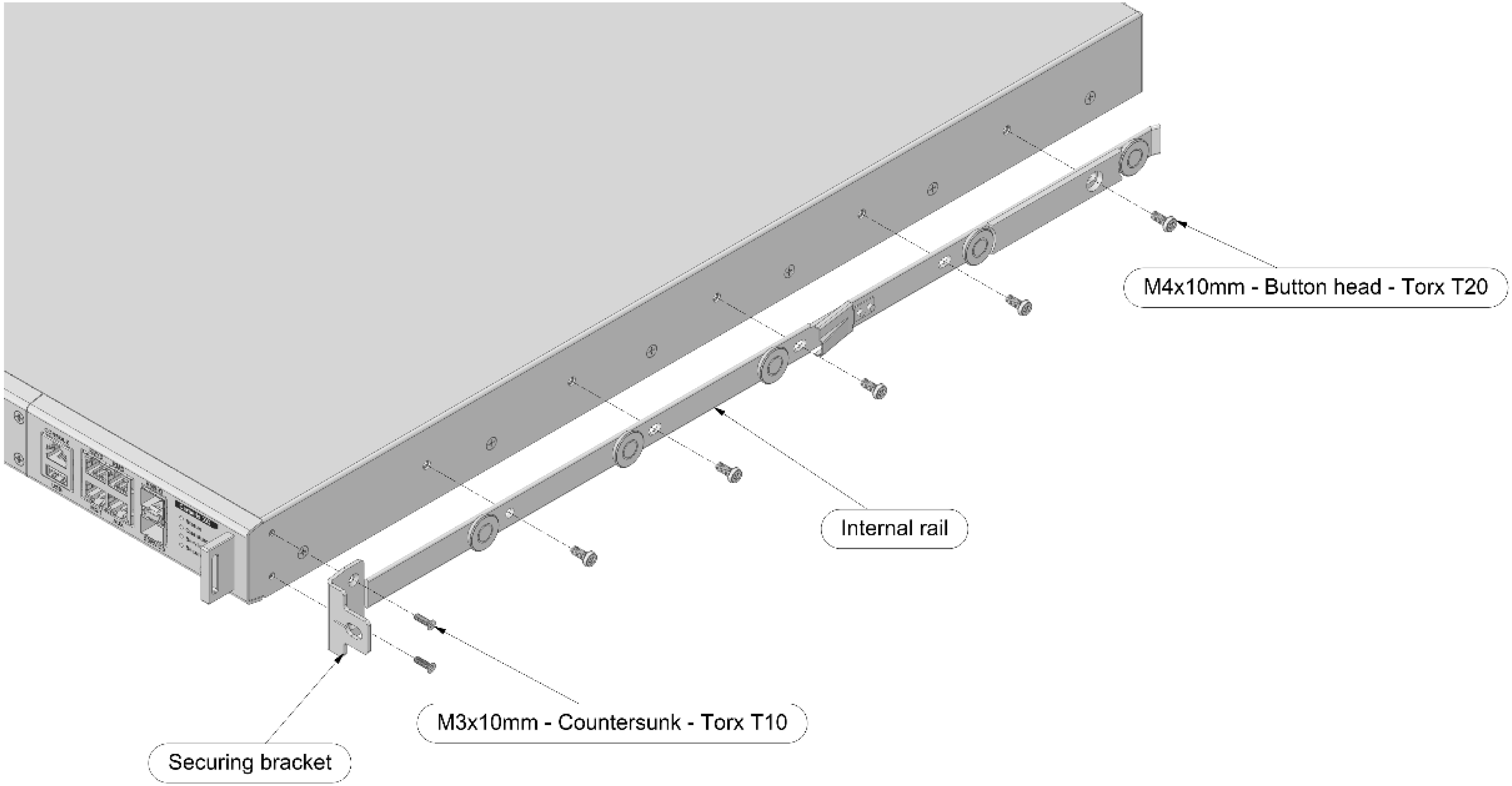
1) Disassemble the internal rails



Cerberis XG – Physical setup (Rack mount)

SWISS QUANTUM

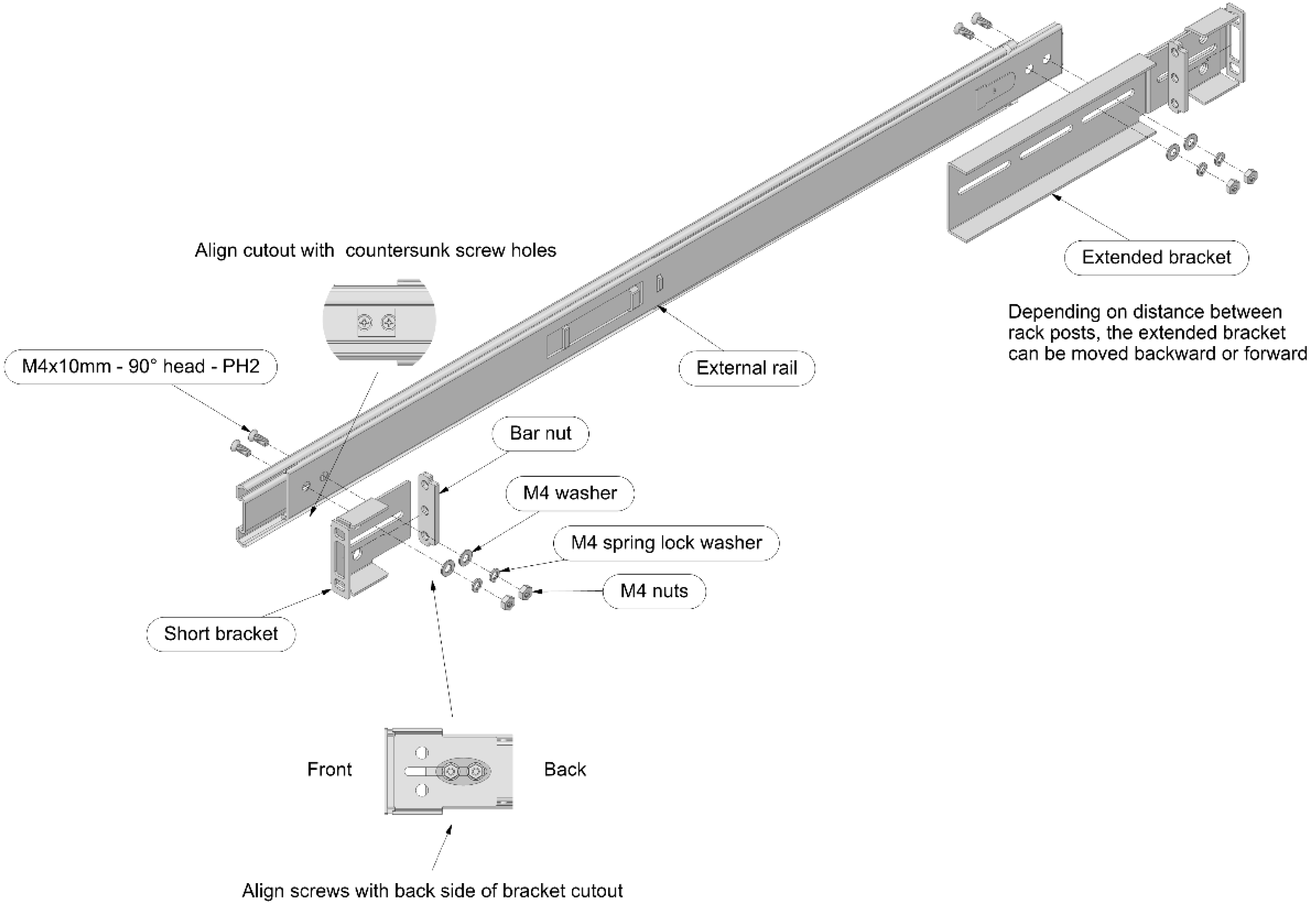
2) Fix the securing brackets and the internal rails to the CerberisXG



IDQ

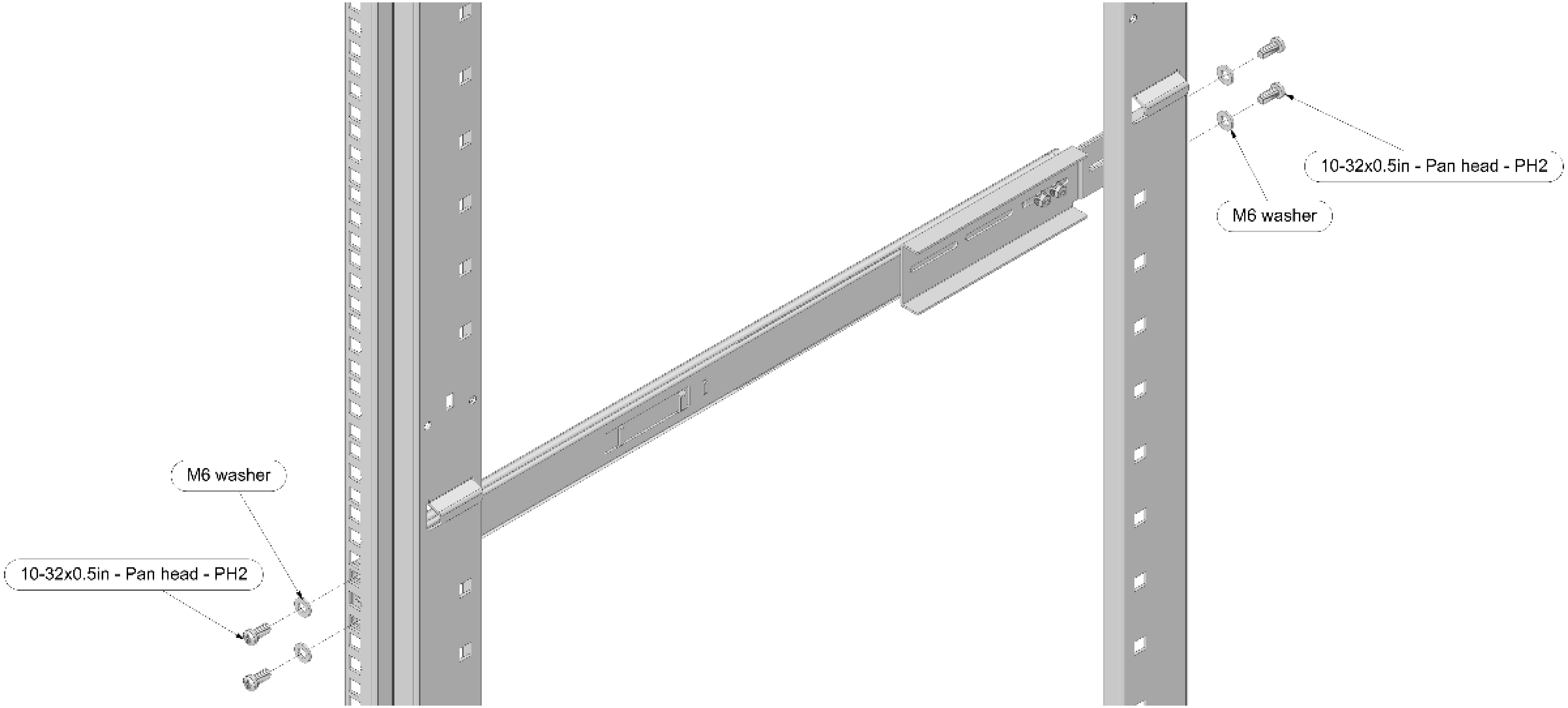
Cerberis XG – Physical setup (Rack mount)

3) Fix the brackets to the external rail. Adjust length by positioning rail in the rack



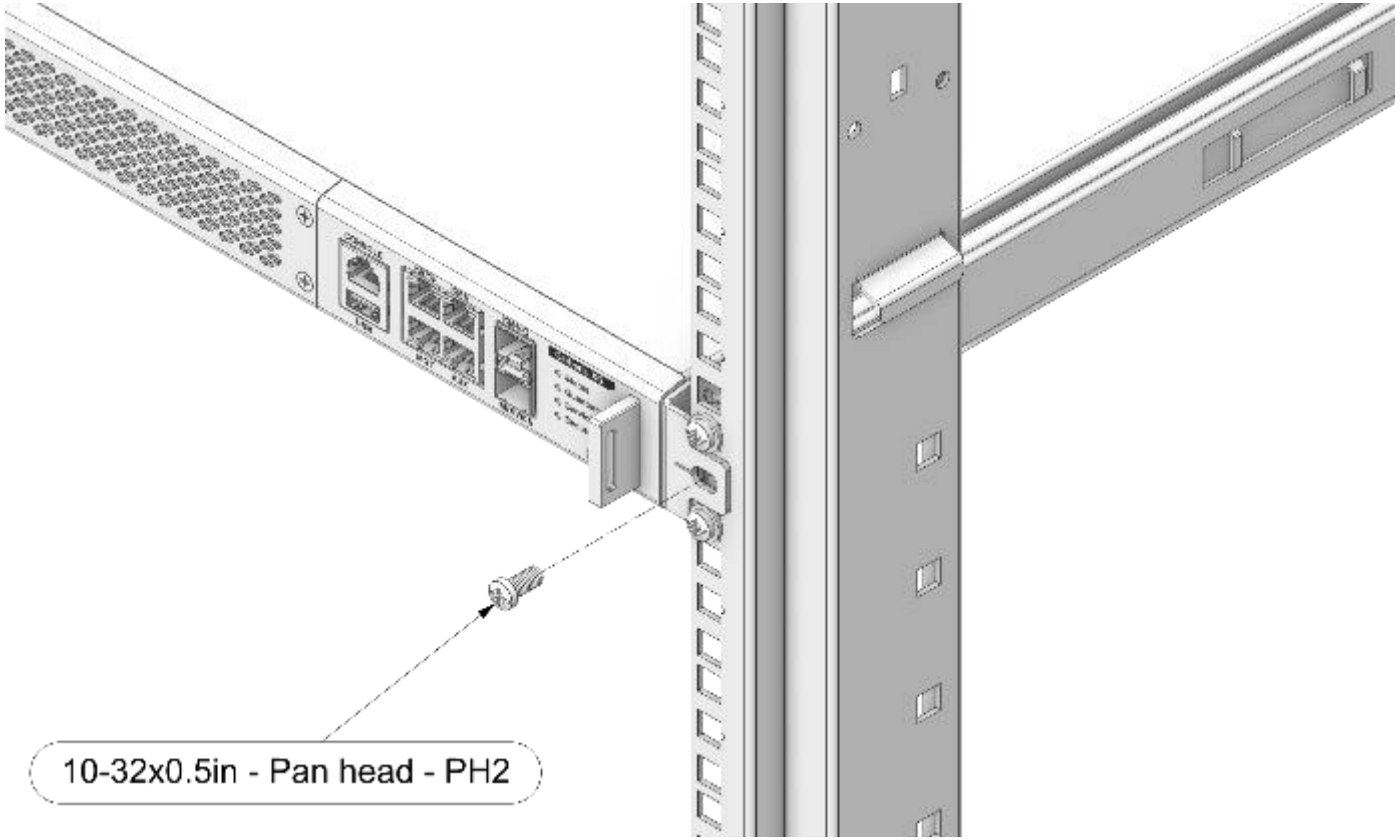
Cerberis XG – Physical setup (Rack mount)

4) Fix the external rails in the rack



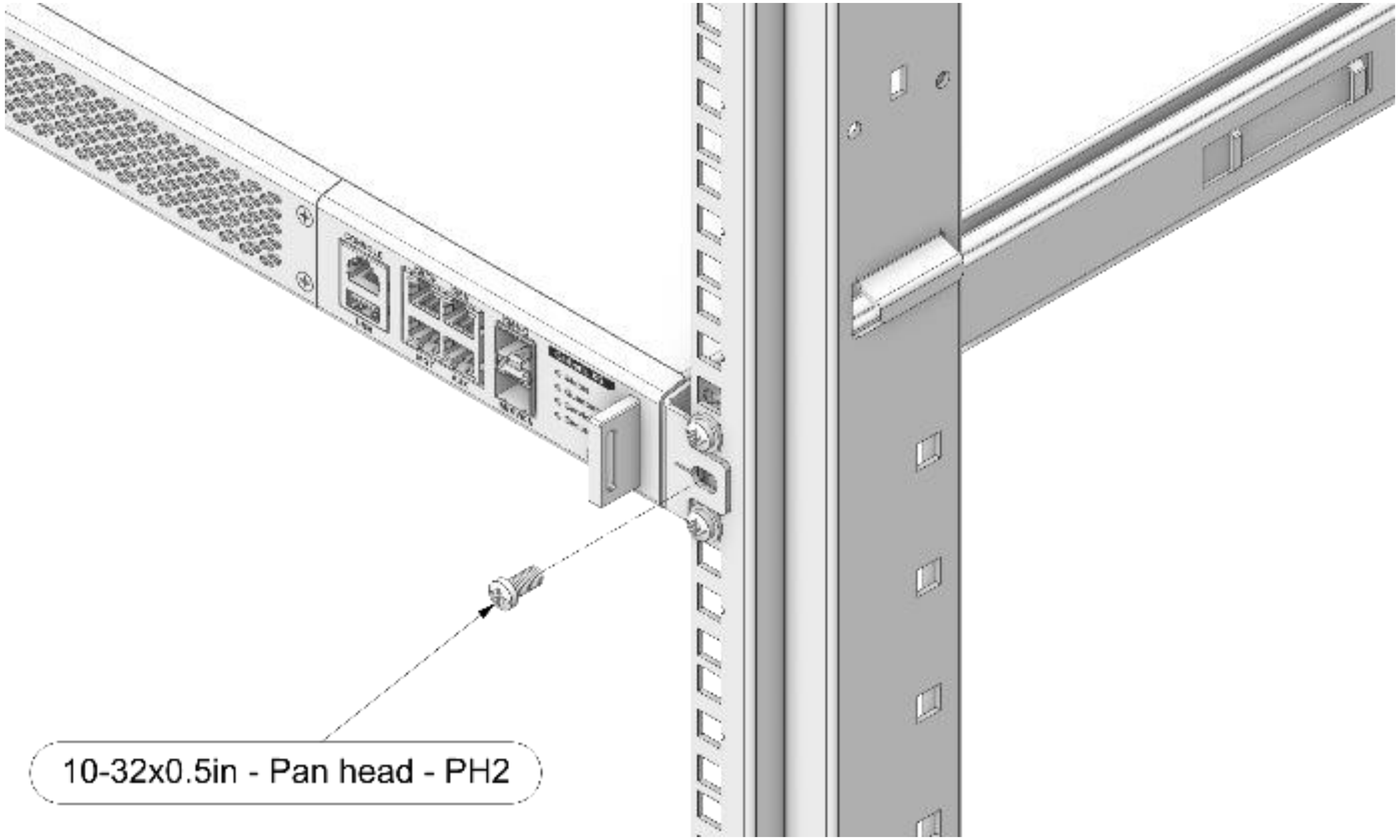
Cerberis XG – Physical setup (Rack mount)

5) Insert carefully the CerberisXG in the external rails and check that it is well secured before you release it.
6) Secure the CerberisXG by fixing the securing brackets.



Cerberis XG – Physical setup (Rack mount)

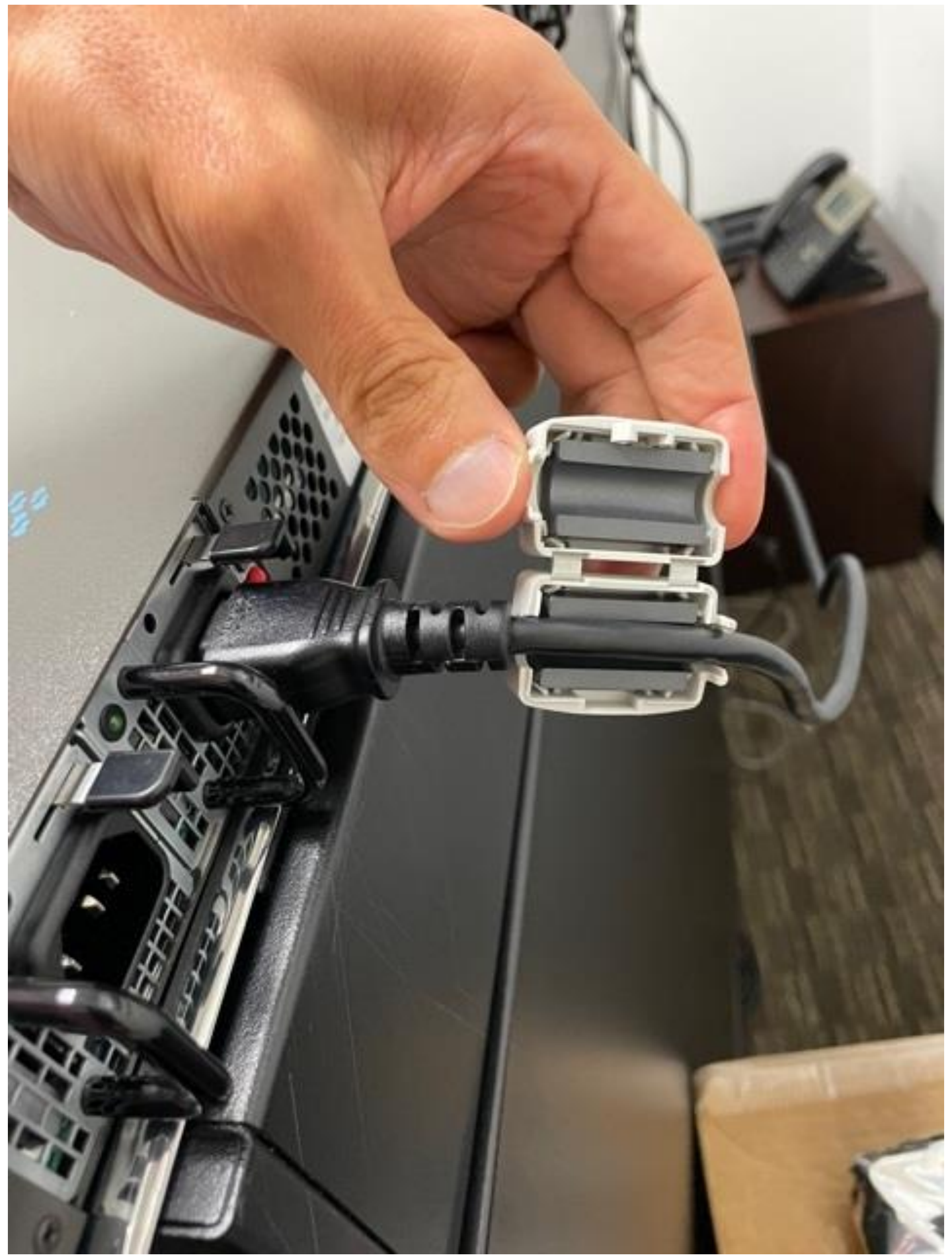
5) Insert carefully the CerberisXG in the external rails and check that it is well secured before you release it.
6) Secure the CerberisXG by fixing the securing brackets.

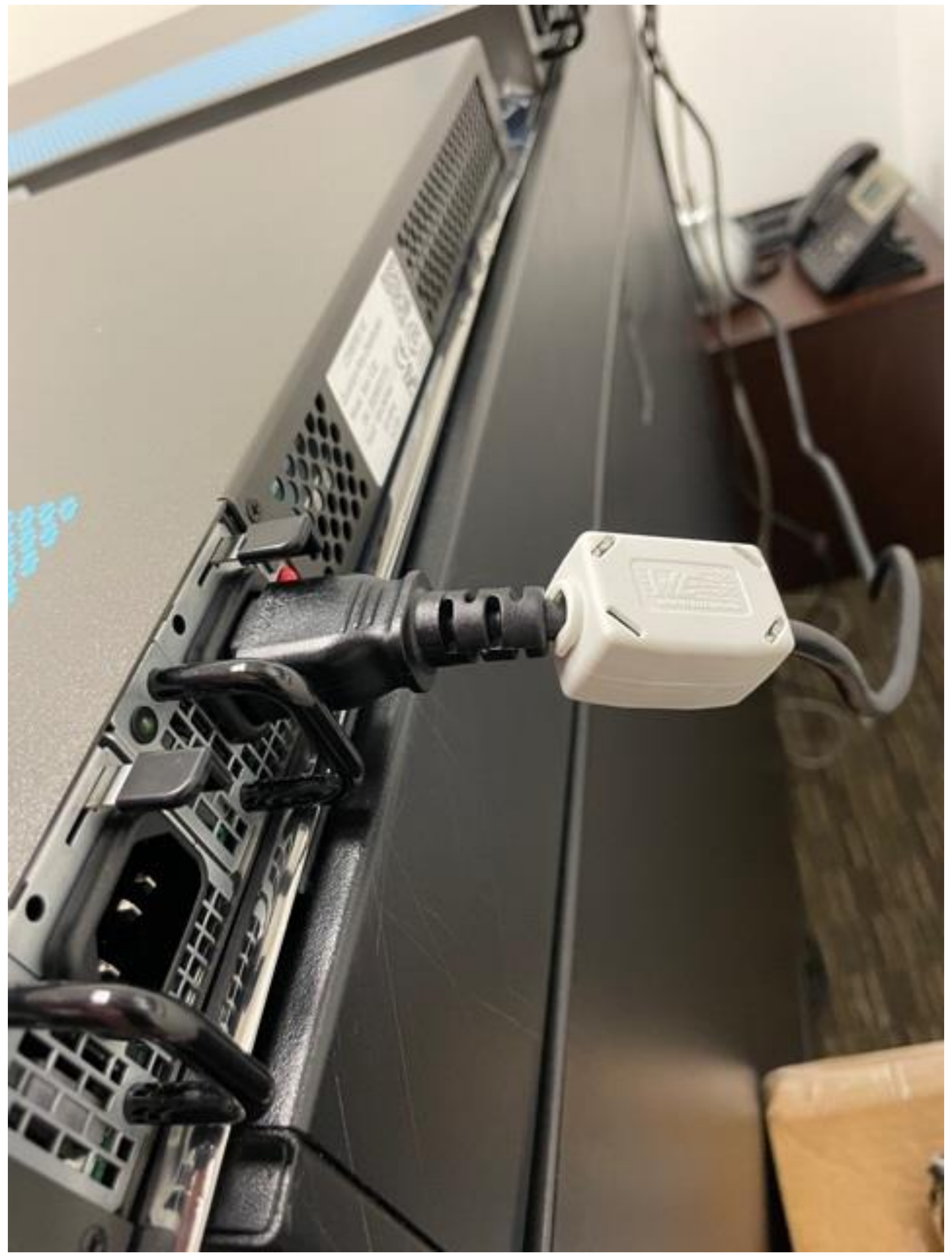


Power Cables

SWISS QUANTUM

Attach the ferrites to the power cables

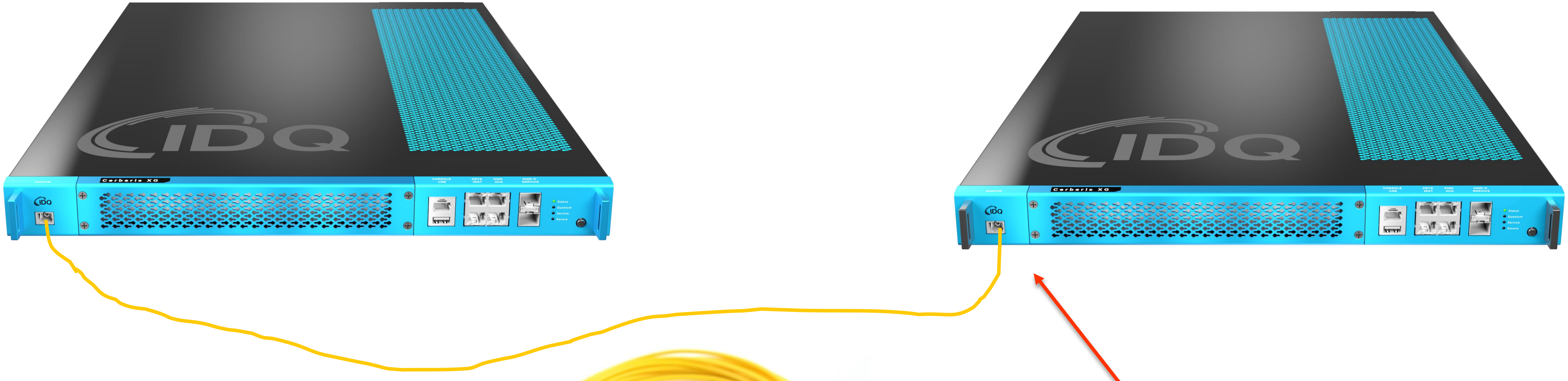




IDQ

SWISS QUANTUM

Connect Quantum Channel



If you are using a short fiber add fixed attenuator

SC/UPC SM fiber

SC/UPC Attenuator (optional)

Quantum Channel Loss

- The Quantum Channel requires a minimum loss of 10dB
- Maximum loss equals the rated budget for your system purchase

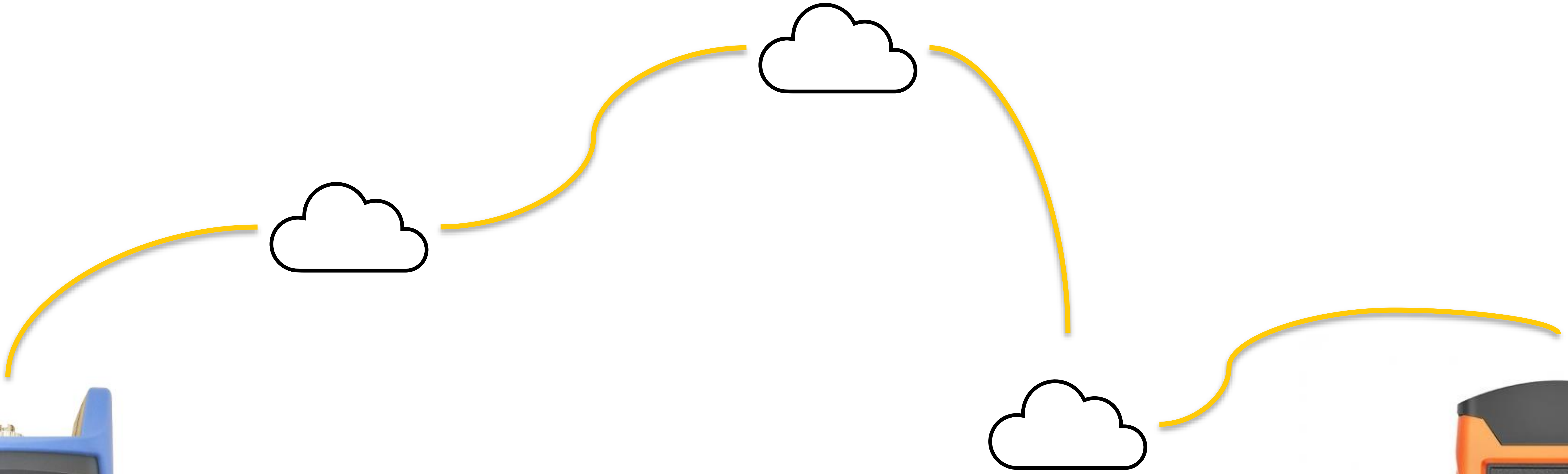
- For Example:
  - A standard 12dB system must have losses between 10dB and 12dB
  - A premium 18dB system must have losses between 10dB and 18dB

- Add the appropriate amount of attenuation as needed for your fiber connection. In a production environment, always measure the link loss using a power meter and laser source… or alternatively use an OTDR.

- If your fiber route has a natural loss of 7dB, add an additional 3dB attenuator to reach the minimum 10dB.

Properly measure Quantum Channel Losses

- Using a laser source and power meter, measure total fiber route including all patches



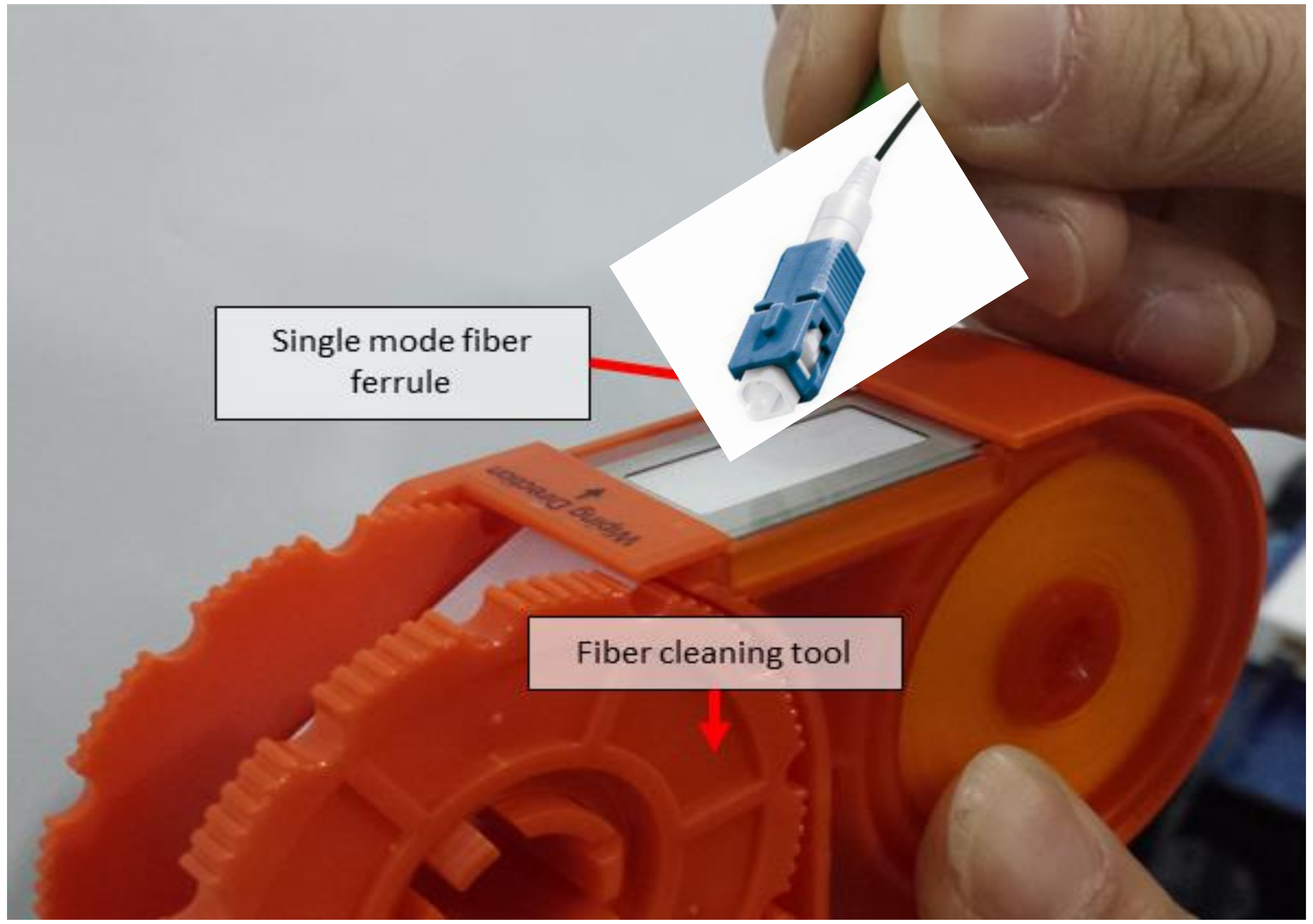
Laser Source
1550/1310nm

If total loss = 7dB, add 3dB or more using a fixed attenuator to the Bob QKD system to reach the minimum 10dB loss

SC/UPC Attenuator

Power Meter

Cerberis XG – Physical setup (Quantum Channel)



- Remove the protective cap from the SC/UPC connector end of the single mode fiber patch cord
- Clean the exposed fiber ferrule end by wiping it twice on the cleaning tool.
- Only wipe in the direction depicted on the cleaning tool!
- Always use a new/clean section of the cleaning tape

Cerberis XG – Physical setup (Quantum Channel)

Attenuator

- Use a SC/UPC connector
- Connect the attenuator if needed
- Make sure the connectors "click" into place and are properly secured.

No Attenuator

Physical Setup – Quantum Channel

SWISS QUANTUM

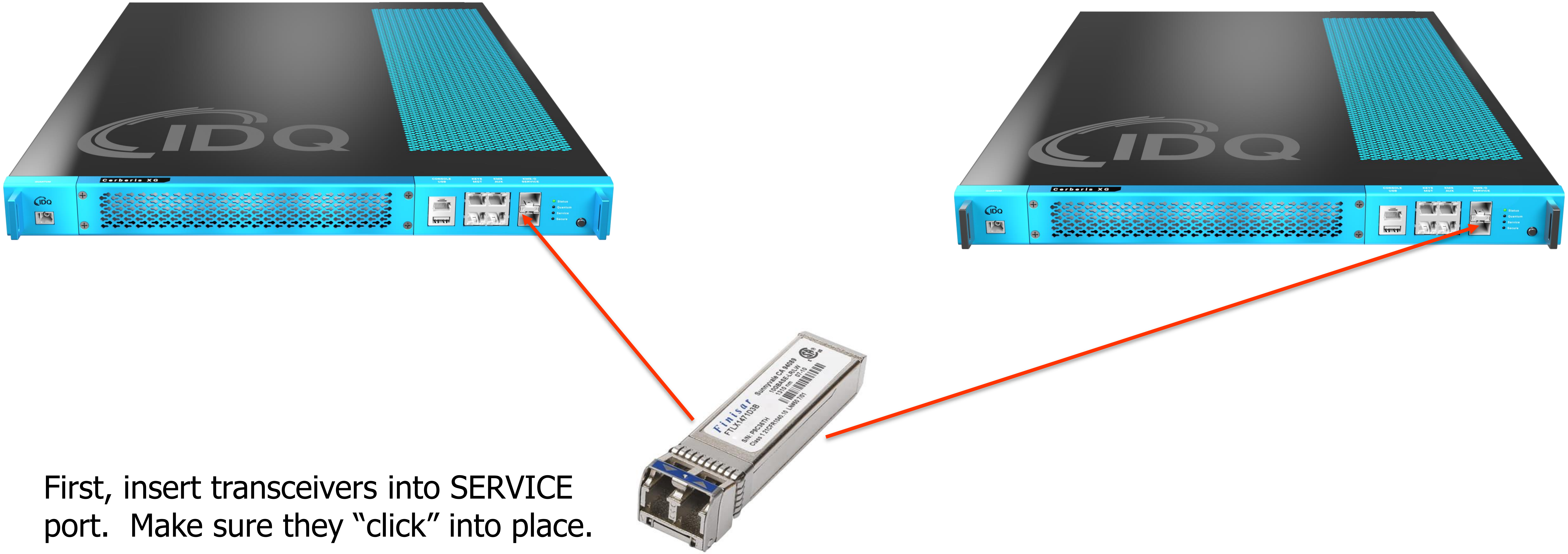
Optical connections, Quantum channel :

1. Remove the protective cap from the SC/UPC connector end of the single mode fiber patch cord and clean the exposed fiber ferrule end by wiping it twice on the cleaning tool. Only wipe in the direction depicted by the arrow.
2. On Alice terminal:
   i. Remove the protective cap on the SC/UPC front panel connector.
   ii. **IMPORTANT:** Verify that the alignment key is well aligned with the alignment slot of the connector!
   iii. carefully connect the fiber to the front panel fiber connector. The connector must clearly “click” in.
3. On Bob terminal if the QC fiber has at least 10dB total losses (real fiber):
   i. Remove the protective cap on the SC/UPC front panel connector.
   ii. IMPORTANT: Verify that the alignment key is well aligned with the alignment slot of the connector!
   iii. carefully connect the fiber to the front panel fiber connector. The connector must clearly “click” in.
4. On Bob terminal if just the 2m SC/UPC is used:
   i. Remove the protective cap on the SC/UPC front panel connector.
   ii. Remove the protective cap from the UPC/SC connector end of the 10dB fix attenuator.
   iii. Remove the protective cap from the SC/UPC connector end of the QC fiber patch cord and clean the exposed fiber ferrule end by wiping it twice on the cleaning tool. Only wipe in the direction depicted by the arrow.
   iv. IMPORTANT: Verify that the alignment key is well aligned with the alignment slot of the connector!
   v. carefully connect the SC/UPC fiber connector to the fix attenuator. The connector must clearly “click” in.
   vi. Remove the protective cap from the SC/UPC connector end of the 10dB fix attenuator. Clean the exposed fiber ferrule end by wiping it twice on the cleaning tool. Only wipe in the direction depicted by the arrow.
   vii. IMPORTANT: Verify that the alignment key is well aligned with the alignment slot of the connector!
   viii. carefully connect the fiber to the front panel fiber connector. The connector must clearly “click” in.

IDQ

Cerberis XG – Physical setup (Service Channel)

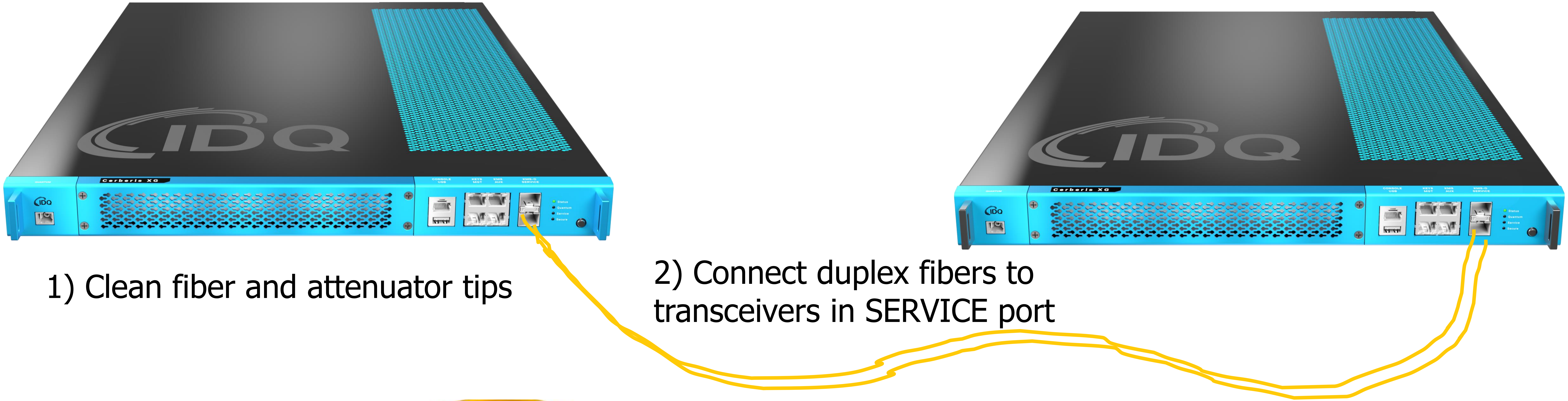
Connect Service Channel



First, insert transceivers into SERVICE port. Make sure they “click” into place.

Cerberis XG – Physical setup (Service Channel)

Connect Service Channel

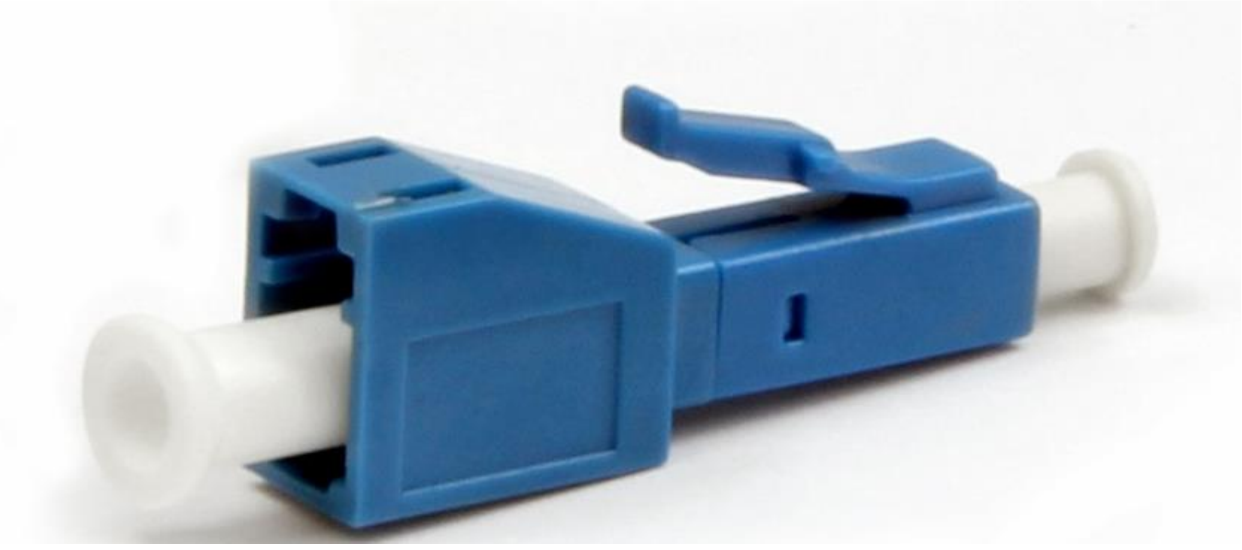


1) Clean fiber and attenuator tips

2) Connect duplex fibers to transceivers in SERVICE port

LC/UPC SMF fiber

+

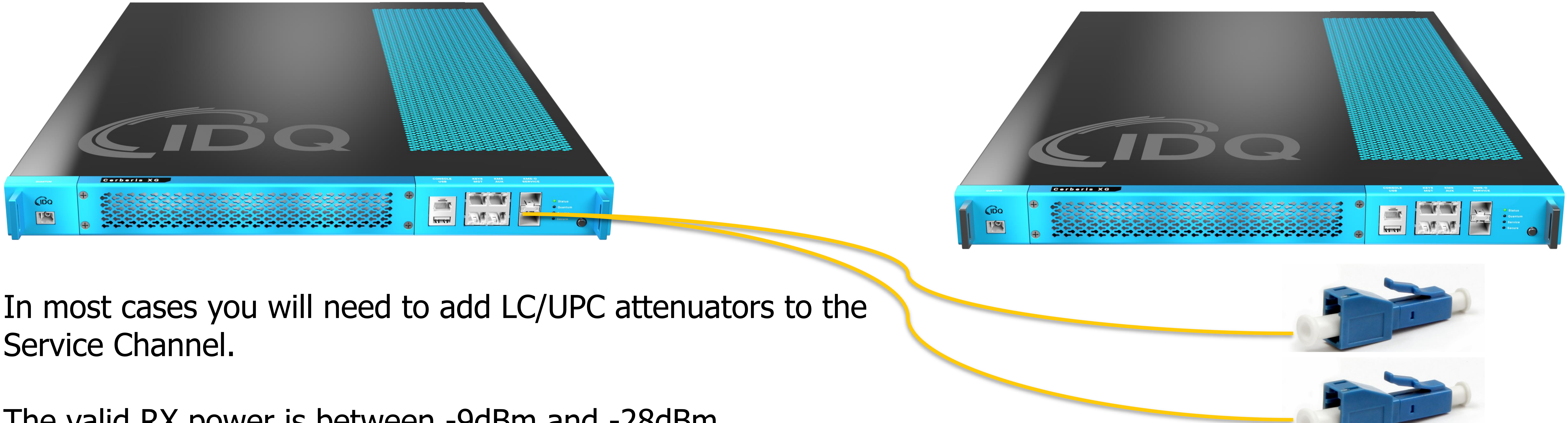


LC/UPC Attenuator (optional)

Cerberis XG – Physical setup (Service Channel)

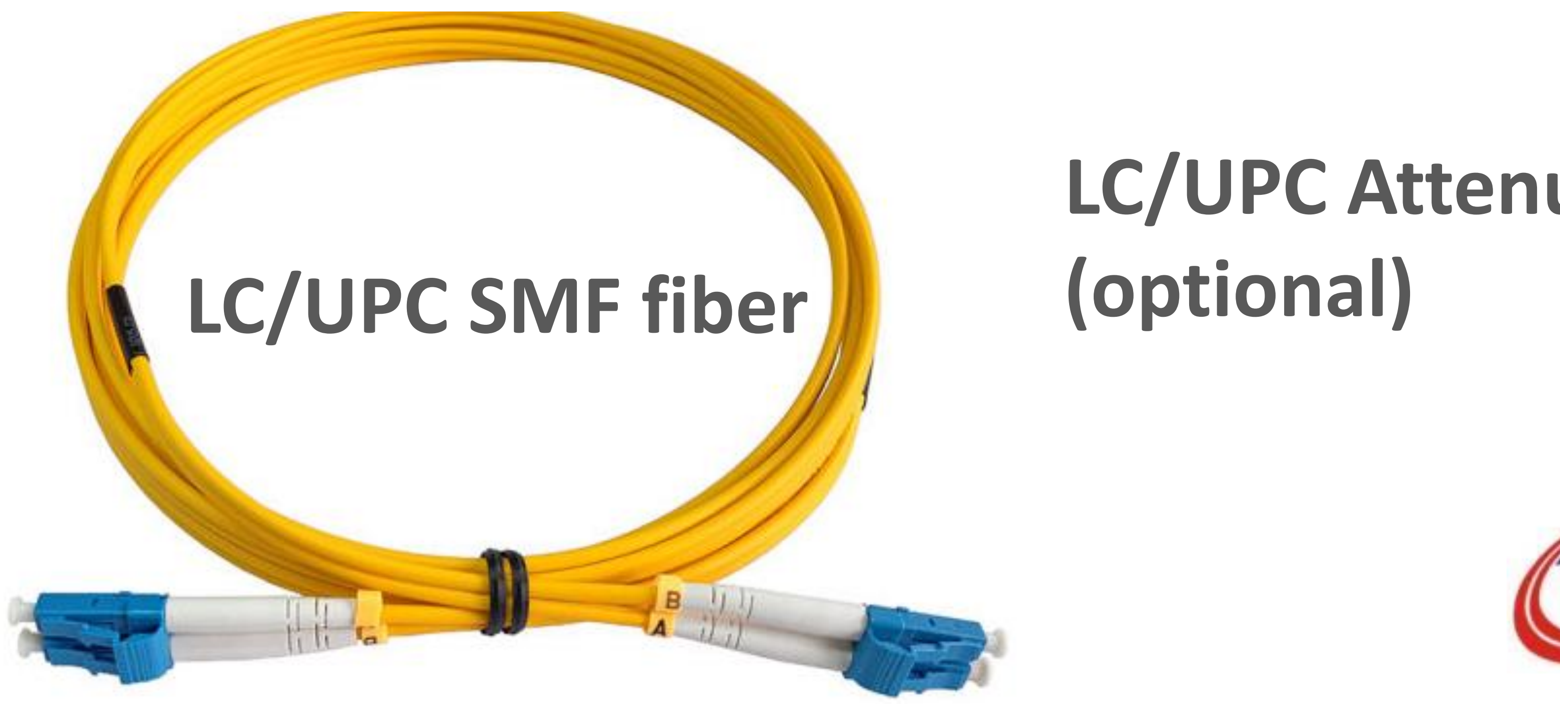
SWISS QUANTUM

Connect Service Channel

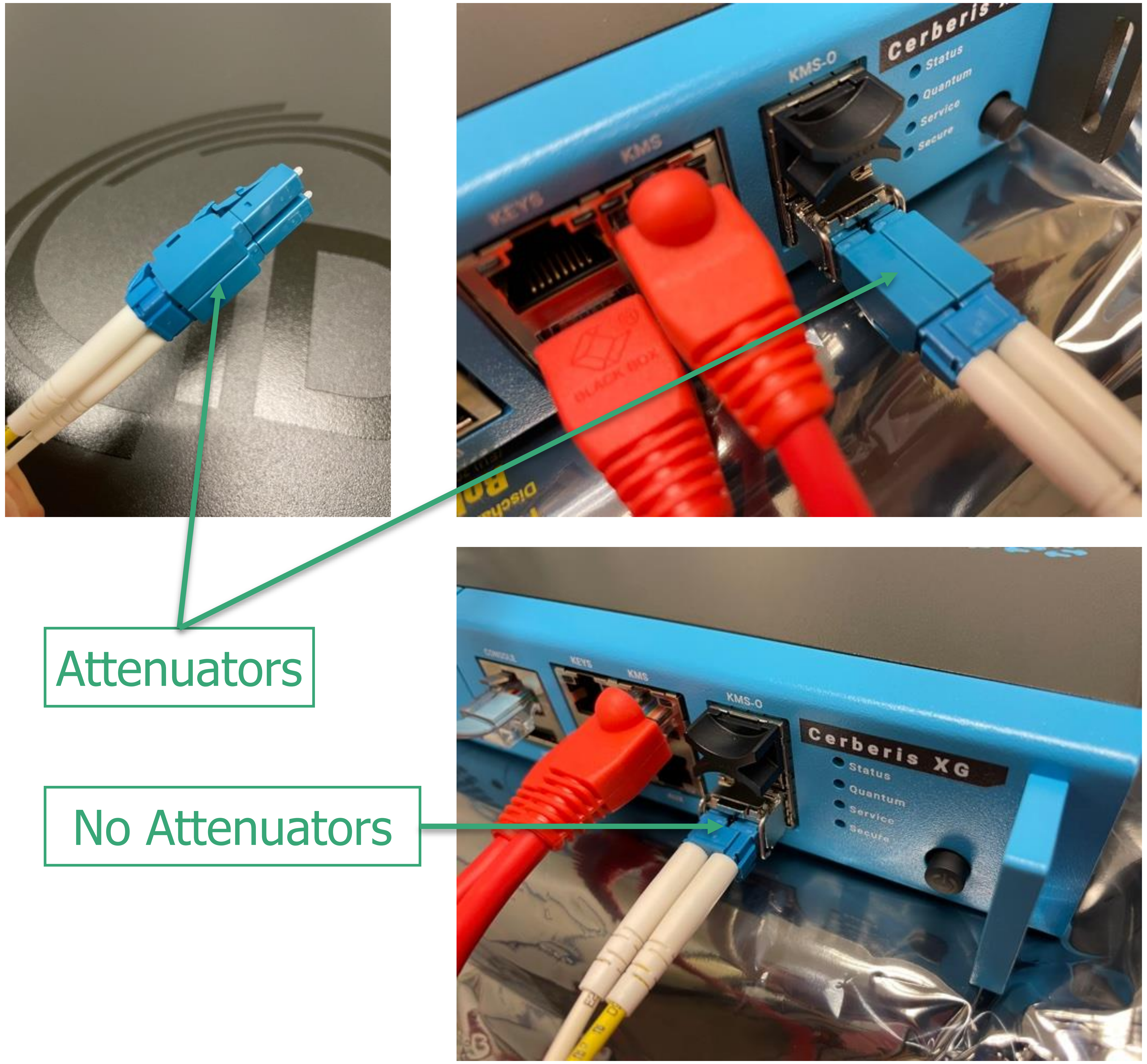


In most cases you will need to add LC/UPC attenuators to the Service Channel.

The valid RX power is between -9dBm and -28dBm
(TX power is approximately +4dBm)
Be sure to measure the RX power to determine necessary losses



Connect Service Channel



- Fixed Channel DWDM 120km SFP Optical Transceiver family.
- Those modules are available for transmitter wavelengths between 1528.77 nm and 1563.86 nm
- Standard system is delivered with SFP ITU Ch30 & Ch29 : 1553.33 & 1554.13

Physical Setup – Service Channel

A. Optical Connections, Service Channel:

1. Remove both protective caps from the LC connector of the dual fiber patch cord and clean the exposed fiber ferrules by wiping them on the cleaning tool. Only wipe in the direction depicted on the cleaning tool!
2. If the service channel fibers have less than 10 dB of losses, add the provided fix LC attenuators to have an attenuation between 10 to 20dB to prevent saturation of the SFP modules.
   i. Remove the protective caps on both sides of the LC fix attenuator and clean the exposed fiber ferrules by wiping them on the cleaning tool. Clean the interior of the LC fix attenuator with the provided cleaning tool.
   ii. Connect the LC/UPC fiber connectors to the fix attenuators.
   IMPORTANT: A proper connection is confirmed by an audible click when inserting the module.
3. Remove the protective cap from the SFP module and carefully insert the LC connector into the SFP module.
4. Slide the SFP module into the front panel mount at the QKD terminal in the slot labelled as Service. A proper installation is confirmed by an audible click when inserting the module.
   IMPORTANT: A proper connection is confirmed by an audible click when inserting the module. Verify that the metallic handle is in a downward position!

Repeat the same procedure for the other terminal (except point 2 if the fix attenuators has been already used).

Installation procedure – Power ON

Details

- Note when QKD system is starting ALICE Service LED is blinking and BOB Service LED is not blinking.
- That means BOB is in the process of cooling down the detectors
- When Quantum LED is green that means the Quantum Channel is working
- When Service LED is green that means the service channel is aligned

Installation procedure – Power ON

LED Representation

LED	LED Status	Status Description
Status	ON (Green)	Cerberis XG node is powered ON
	OFF	Cerberis XG node is powered OFF
Quantum	ON (Orange)	Cerberis XG node Quantum Channel Error
	ON (Green)	Cerberis XG node Quantum Channel OK
Service	Blinking (Orange)	Cerberis XG node Service Channel is desynchronized
	ON (Orange)	Cerberis XG node Service Channel is synchronizing
	ON (Green)	Cerberis XG node Service Channel is synchronized

Cerberis XG – Console cables

The QKD is delivered with console cables.

They are mandatory to do the basic configuration.

Cerberis XG – Physical setup (Console)

- **Preliminary configuration done with the CLI console port**

Alternatively use SSH to the default IP addresses

Configuration (CLI)

- You must use the serial console cables
- Speed bauds: 115200
- Date bits: 8 / Stop bits: 1 / Parity: none
- Flow control: XON/XOFF

CERBERIS XG

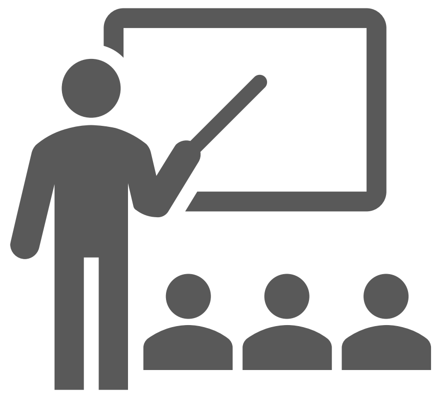
Preliminary configuration

Cerberis XG – Preliminary configuration

SWISS QUANTUM

Goal of this part

- Understand how to connect to the elements using serial
- Be able to understand the necessary IP addresses
- Be able to configure the QKD system



Cerberis XG – Preliminary configuration

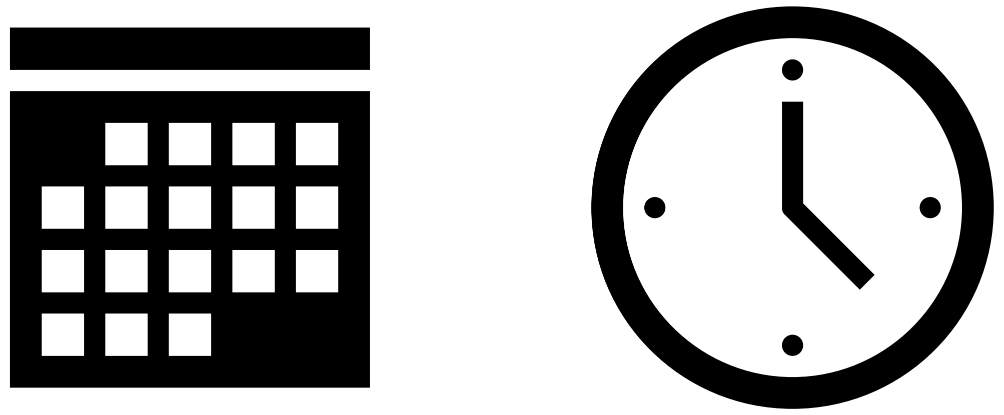
Management:

- **All pre-configuration** is done **through CLI** using
  - **SSH (port 22)**
  - **Serial Console connection**
- **There is no GUI** or graphical interfaces for this part
- **Health and status** of the system can be checked using
  - **Logs**
  - **SNMPv3**
  - **QMS/QNET – After initial configuration**
- **Logs** can be forwarded to a syslog server
- Note if using putty on CentOS you may need to run putty in sudo mode to access the USB/Serial interface port.
- If you enter an incorrect login password 3x you will be locked out without warning. A reboot is required to correct this.

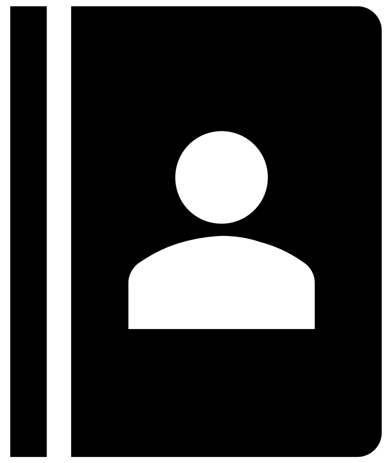
Cerberis XG – Preliminary Configuration

SWISS QUANTUM

- Date & Time
  - It is necessary to configure the date and time for proper topology synchronization and network certificate use
  - Consider configuring an NTP server

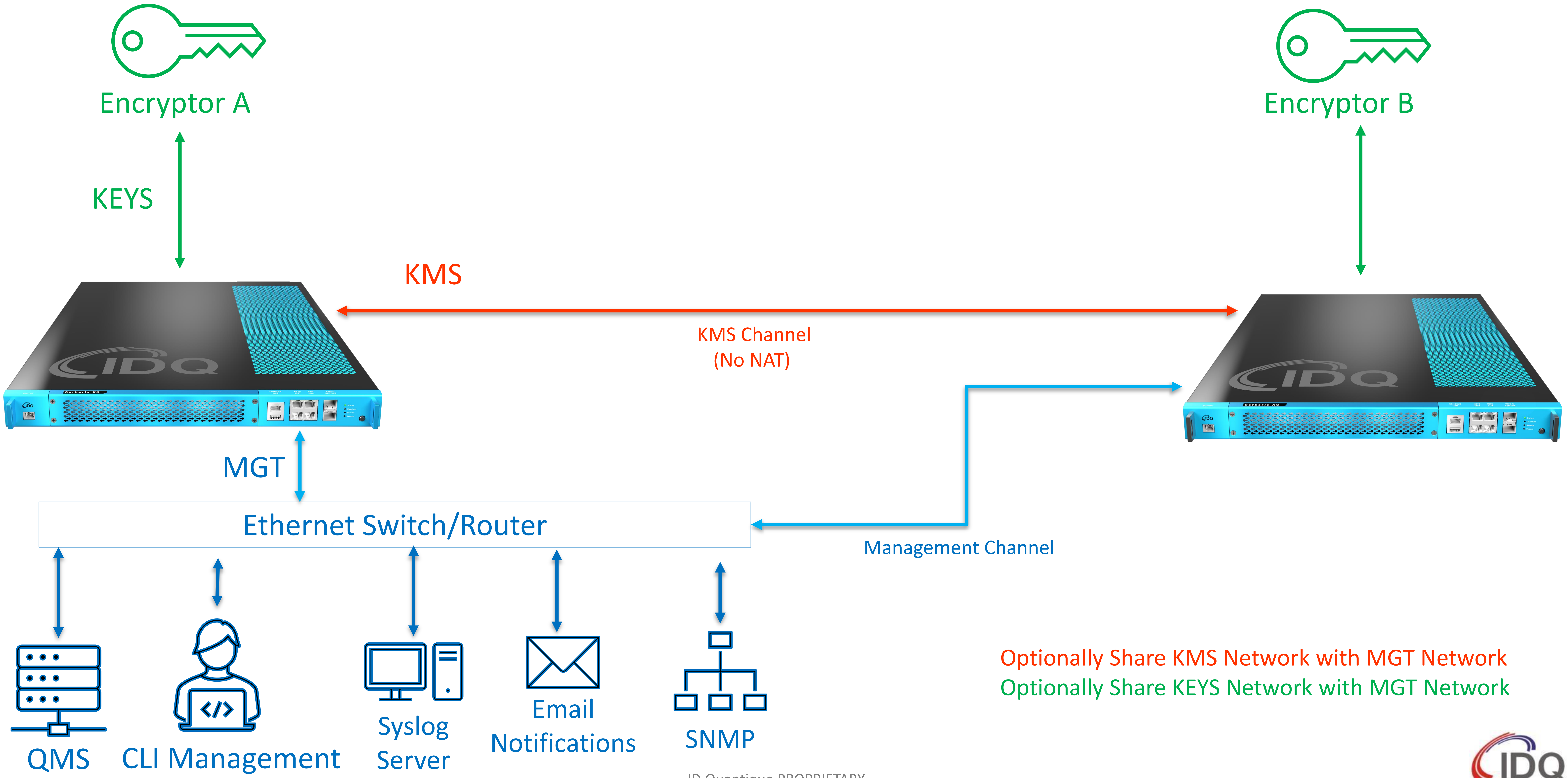


- IP Address Plan
  - Understand the necessary IP addresses
  - Choose the necessary IP addresses
  - Configure the MGT IP address
  - Configure the KMS IP address
  - Configure the KEYS IP address



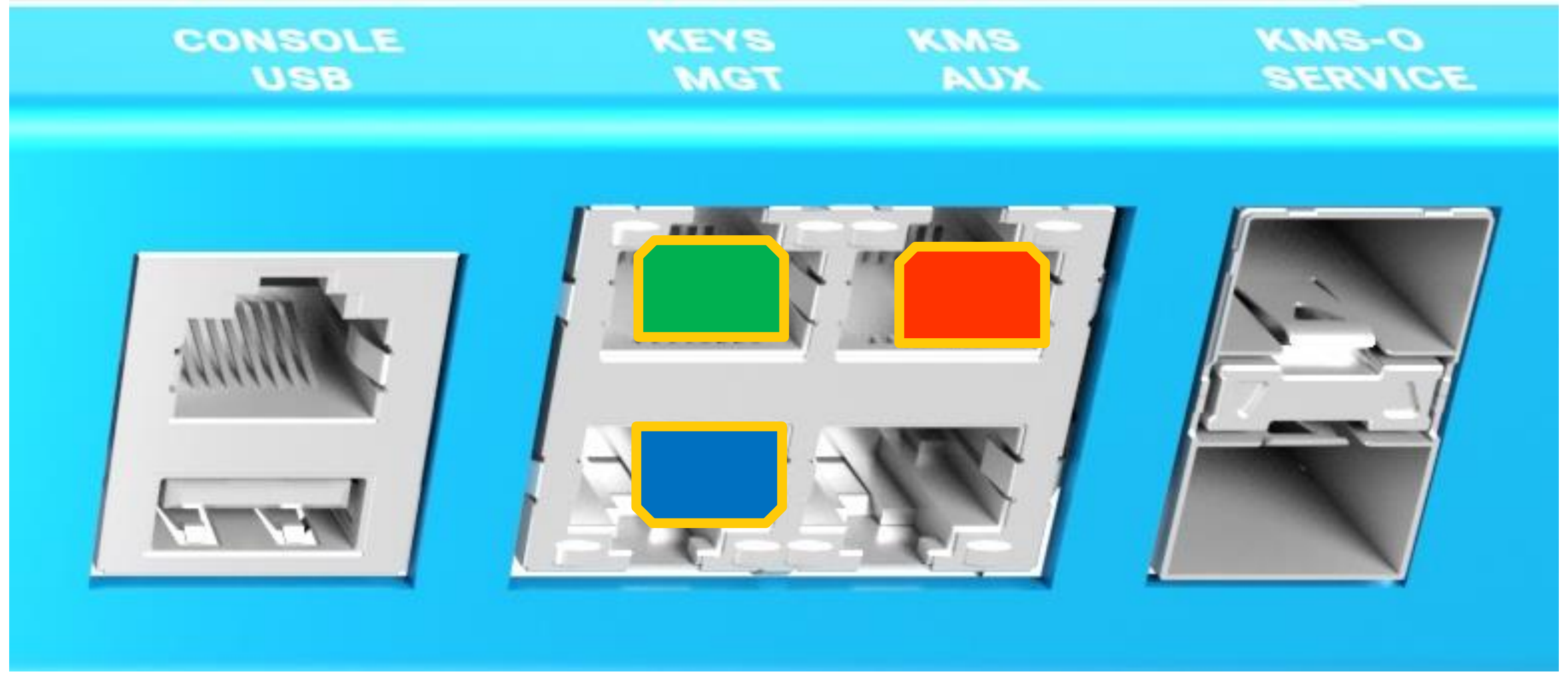
x.x.x.x

IP Addressing Considerations



IP Addressing Considerations

- The KEYS Interface needs to communicate with the encryptors (consumer). This interface uses the ETSI or similar key exchange protocol
  - Optionally share MGT interface.
- The KMS Interface on Alice needs to communicate with the KMS Interface on Bob. You cannot perform NAT on this connection. All KMS's in the QKD topology need to communicate.
  - Optionally share MGT interface – Still cannot NAT the KMS traffic
- The MGT Interface needs to communicate with the remote QKD nodes, your management PC, QMS Topology Manager, Syslog Server, SNMP Tool, Email Server
  - Optionally this interface can be used for KMS and KEY Exchange as well.



LAB IP addresses Example

10.10.1.xxx QMS SERVER

10.10.1.xxx MGT ALICE
192.168.xxx.xxx KEYS ALICE
10.10.2.xxx KMS ALICE
10.10.3.xxx AUX ALICE

10.10.1.xxx MGT BOB
192.168.xxx.xxx KEYS BOB
10.10.2.xxx KMS BOB
10.10.3.xxx AUX BOB

Mask 255.255.xxx.xxx
Gateway 10.10.xxx.xxx

Needed IP addresses on the Cerberis3 systems

LAB IP addresses Example

10.10.xxx.xxx QMS SERVER

10.10.xxx.xxx QNC ALICE
10.10.xxx.xxx QKD ALICE
10.10.xxx.xxx Shelf Manager Alice bound (used for upgrade)
10.10.xxx.xxx Switch Management Alice (used for management)
192.168.xxx.xxx ETSI frontpane p2 ALICE

10.10.xxx.xxx QNC BOB
10.10.xxx.xxx QKD BOB
10.10.xxx.xxx Shelf Manager BOB bound (used for upgrade)
10.10.xxx.xxx Switch Management BOB (used for management)
192.168.xxx.xxx ETSI frontpane p2 BOB

Mask 255.255.xxx.xxx
Gateway 10.10.xxx.xxx

Cerberis XG – Preliminary configuration

Connect to the CerberisXG via the CONSOLE Port
Alternatively, SSH to the device using the Default IP address:

192.168.10.102 (Alice)
192.168.10.107 (Bob)

Default Credentials:

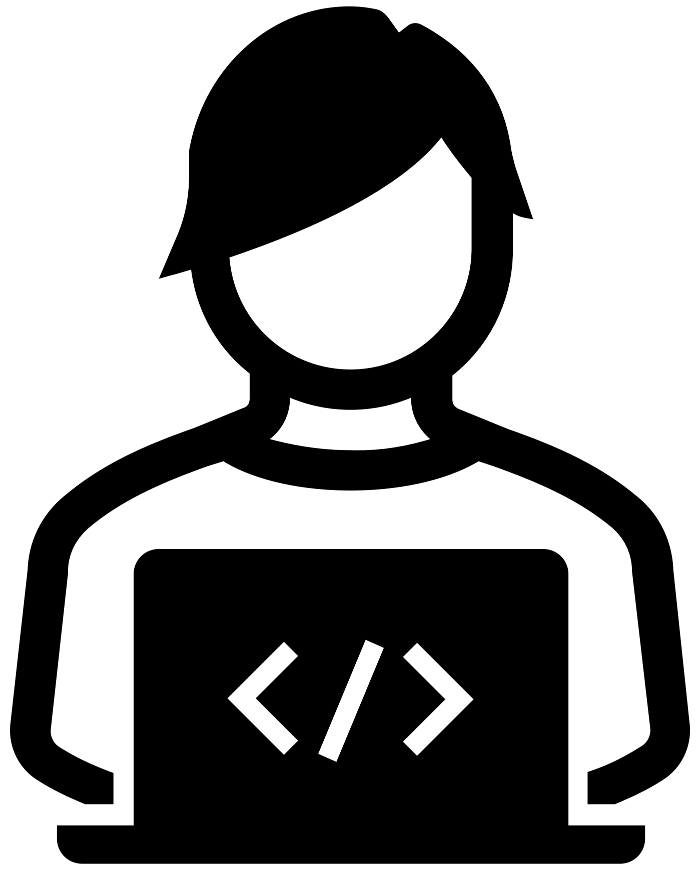
Username: admin
Password: admin

Type

“date --help” to change the date/time
Change the time zone before changing the time
(Alternatively consider configuring NTP server)

Type

“network -i 0” to change the MGT IP address ← Mandatory
“network -i 1” to change the KEYS IP address
“network -i 2” to change the AUX IP address ← Mandatory… even if you are not using this AUX interface!
“network -i 3” to change the KMS IP address



Cerberis XG - Initial Login

1. Login as “admin”:
   > admin
   default password: admin
2. The first time the system is accessed as admin a password change is requested.

NOTE: Before setting the new password, it is important to mention that the new password must fulfil the following criteria:

- Minimum password length is 15 characters.
- No character can be repeated more than twice.
- Password must be composed of capital/small letters, numbers, and special characters.
- Have maximum 4 consecutive characters of the same class.
- A new password must differ of 8 characters at least from the old password.

Cerberis XG – Preliminary configuration

```
admin> date --help
manage the datetime and timezone configuration

Usage: date [command] [options]

Options:
  --help  Show help information.

Commands:
  set     manage the datetime configuration
  set-tz  manage the timezone configuration

Run 'date [command] --help' for more information about a command.

admin> date set-tz
Please identify a location so that time zone rules can be set correctly.
Please select a continent or ocean.
1) Africa            4) Arctic Ocean     7) Australia       10) Pacific Ocean
2) Americas          5) Asia             8) Europe
3) Antarctica        6) Atlantic Ocean   9) Indian Ocean
#?
```

Help

Command to configure timezone

```
admin> date set --help
manage the datetime configuration

Usage: date set [options] <datetime>

Arguments:
  datetime  Supported formats [yyyy-mm-ddThh:mm:ss | yyyy-mm-dd | hh:mm:ss]

Options:
  --help    Show help information.

admin> date set 2021-02-09T12:26:00
2021-02-09 12:26:00 -05:00
Tue Feb  9 12:26:05 EST 2021
admin>
```

Help

Command to configure date and time

Cerberis XG – Preliminary configuration

SWISS QUANTUM

```
admin> network --help
manage the network configuration

Usage: network [command] [options]

Options:
  --help                  Show help information.
  -i|--interface <arg>    Set the network interface (index)
  -a|--addr <arg>         Set the IP address
  -n|--netmask <arg>      Set the netmask
  -g|--gateway <arg>      Set the gateway
  --dns1 <arg>            Set the DNS1
  --no-dns1               Remove the DNS1
  --dns2 <arg>            Set the DNS2
  --no-dns2               Remove the DNS2
  -m|--mode <arg>         Set the port mode (fiber or copper)
                          Allowed values are: COPPER, FIBER.
  -f|--force              Don't ask for confirmation

Commands:
  hostname                show or change the host name
  test                    test network

Run 'network [command] --help' for more information about a command.

admin> network -i 0 -a 192.168.10.101 -n 255.255.255.0 -g 0.0.0.0
The CLI connection will be broken and the network will be restarted.
You will need to reconnect to the CLI with the configuration:
IpAddress: 192.168.10.101
Netmask:   255.255.255.0
Gateway:   0.0.0.0
Are you sure you want to continue? [Y/n]
```

Help

Set CerberisXG/QNC management IP address

Cerberis XG – Preliminary configuration

```
admin> network
Interface 0
-----------
IPAddress: 10.10.10.102
Netmask  : 255.0.0.0
Gateway  : 0.0.0.0

Interface 1
-----------
IPAddress: 192.168.10.1
Netmask  : 255.255.255.0
Gateway  : 0.0.0.0

Interface 2
-----------
IPAddress: 192.168.20.1
Netmask  : 255.255.255.0
Gateway  : 0.0.0.0

Interface 3
-----------
IPAddress: 192.168.30.1
Netmask  : 255.255.255.0
Gateway  : 0.0.0.0

admin>
```

Confirm Settings

MGT

KEY

AUX

KMS

Cerberis3 – Preliminary configuration

```
admin> network
Interface 0
-----------
IPAddress: 192.168.10.101
Netmask  : 255.255.255.0
Gateway  : 0.0.0.0

Interface 2
-----------
IPAddress: 192.168.20.2
Netmask  : 255.255.255.252
Gateway  : 0.0.0.0

Interface 3
-----------
IPAddress: 192.168.30.3
Netmask  : 255.255.255.252
Gateway  : 0.0.0.0

admin>
```

Confirm Settings

Optional QNC Frontpane Key Delivery Interfaces

Cerberis 3 – Preliminary configuration

```
admin> network --help
manage the network configuration

Usage: network [command] [options]

Options:
  --help                  Show help information.
  -i|--interface <arg>    Set the network interface (index)
  -a|--addr <arg>         Set the IP address
  -n|--netmask <arg>      Set the netmask
  -g|--gateway <arg>      Set the gateway
  --dns1 <arg>            Set the DNS1
  --no-dns1               Remove the DNS1
  --dns2 <arg>            Set the DNS2
  --no-dns2               Remove the DNS2
  -m|--mode <arg>         Set the port mode (fiber or copper)
                          Allowed values are: COPPER, FIBER.
  -f|--force              Don't ask for confirmation

Commands:
  hostname                show or change the host name
  test                    test network

Run 'network [command] --help' for more information about a command.

admin> network -i 2 -a 192.168.20.2 -n 255.255.255.0 -g 192.168.20.1
The CLI connection will be broken and the network will be restarted.
You will need to reconnect to the CLI with the configuration:
IpAddress: 192.168.20.2
Netmask:   255.255.255.0
Gateway:   192.168.20.1
Are you sure you want to continue? [Y/n]
```

Configure the Frontpane port using the “network” command:

Help and command’s information

Command to use with arguments

Example: encryptor could use IP 192.168.20.20
You can choose between port 2 or 3 with the –i argument

Cerberis 3 – Preliminary configuration

SWISS QUANTUM

```
admin> network --help
manage the network configuration

Usage: network [command] [options]

Options:
  --help                  Show help information.
  -i|--interface <arg>    Set the network interface (index)
  -a|--addr <arg>         Set the IP address
  -n|--netmask <arg>      Set the netmask
  -g|--gateway <arg>      Set the gateway
  --dns1 <arg>            Set the DNS1
  --no-dns1               Remove the DNS1
  --dns2 <arg>            Set the DNS2
  --no-dns2               Remove the DNS2
  -m|--mode <arg>         Set the port mode (fiber or copper)
                          Allowed values are: COPPER, FIBER.
  -f|--force              Don't ask for confirmation

Commands:
  hostname                show current host name
  set-hostname            change the host name

Run 'network [command] --help' for more information about a command.

admin> network -i 0 -a 192.168.10.102 -n 255.255.255.0 -g 192.168.10.1
The CLI connection will be broken and the network will be restarted.
You will need to reconnect to the CLI with the configuration:
IpAddress: 192.168.10.102
Netmask:   255.255.255.0
Gateway:   192.168.10.1
Are you sure you want to continue? [Y/n]
```

Admin role must be used for these changes

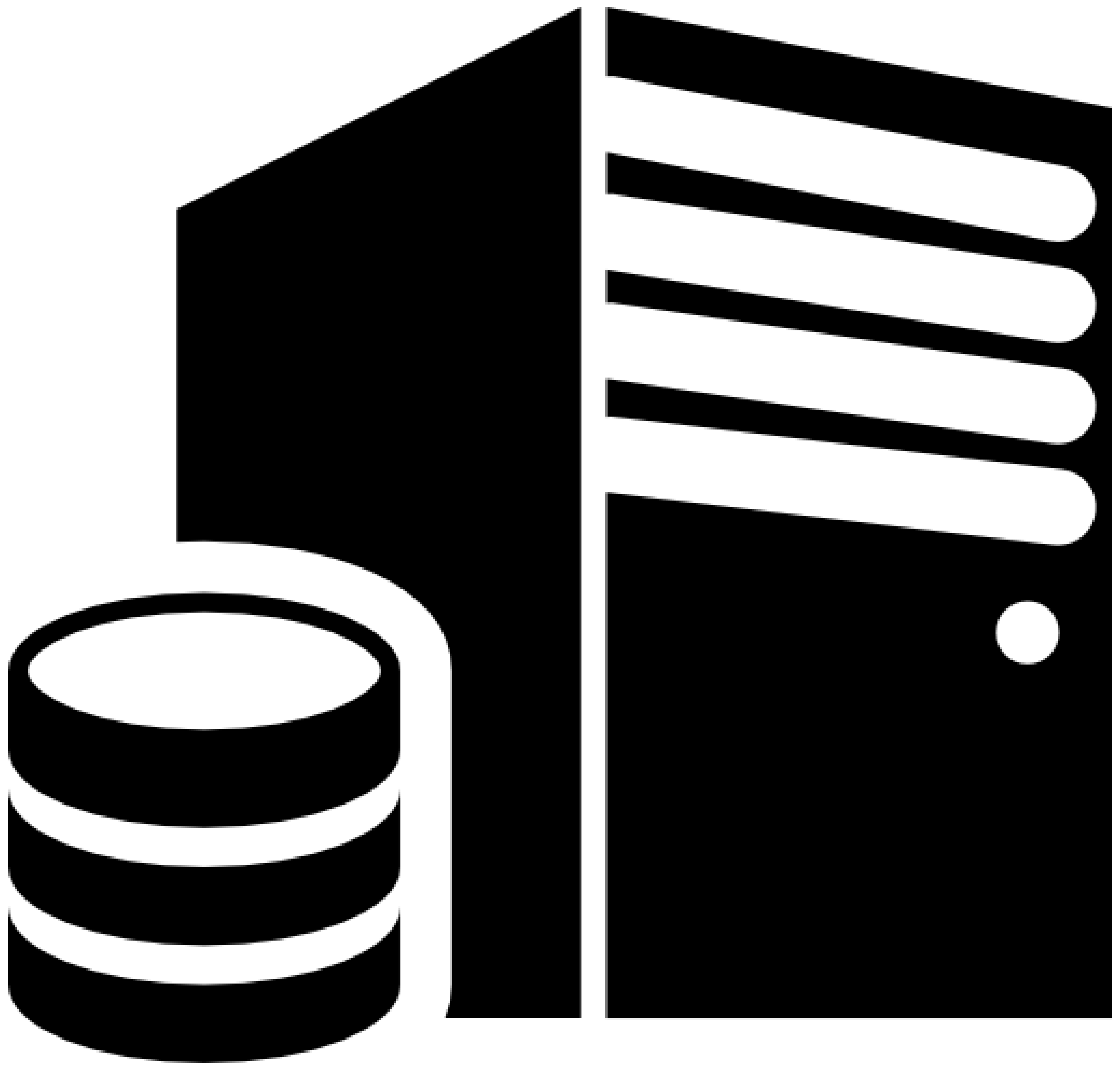
Help and command's information

Command to use with arguments

Type "network" to confirm changes

IDQ

QNET & QMS installation



Migration from a Previous Version:

If you are migrating from a previous version, or migrating from KEMS to QMS, please do the following:

Export the current KEMS/QMS configuration using QNET:

qnet export group –o <filename> <group name>

Shut down the old KEMS/QMS Docker container from the installation directory:

sudo docker-compose down

(Optionally destroy all old Docker containers using "sudo docker system prune" / "sudo docker system prune –a")

Perform the QKD system and QNET/QMS software upgrade procedure

Import the old configuration using:

qnet import group <group name> <file name>

<u>ALTERNATIVELY</u>:
The current QMS supports auto-migration from one QMS version to the next.
Copy the new QMS tar file to the same directory which contains the QMS folder.
Decompress the new QMS package as per the installation instructions. It should maintain the existing configuration.

QMS installation - Docker

Docker installation

- Follow the installation instructions provided by Docker
- Currently QMS is only supported on Ubuntu20 – There are known issues with CentOS.
- https://docs.docker.com/engine/install/ubuntu/
- Recommended to install latest stable version of docker and docker-compose

- Ubuntu20 is the recommended Operating System for QMS/QNET Installation

Docker Installation

Pre-requisite

a. QNET Central Server minimum configuration:
   i. Linux Ubuntu20 preferred (The latest supported Ubuntu release is: Ubuntu 20.04 (not .x later))
   ii. Disk: 100 GB
   iii. CPU: 4 Cores (8 Cores for complex QKD configurations)
   iv. RAM: 16GB (32 GB for complex QKD configurations)

Uninstall older versions of Docker and Docker-Compose

Docker Installation

In a fresh installation of Ubuntu20, typing “Docker” in a terminal window should provide the recommended command to install Docker.

(typically something like “sudo apt install docker”

If docker doesn’t start automatically after system reboot, use the following command to start the docker daemon:

```
# sudo systemctl start docker
```

Docker Installation

vi. Verify that Docker Engine is installed correctly by running the hello-world image

```
# sudo docker run hello-world
```

This command downloads a test image and runs it in a container. When the container runs, it prints an informational message and exits.

Docker Engine is installed and running. You need to use sudo to run Docker commands. Continue to Linux postinstall to allow non-privileged users to run Docker commands and for other optional configuration steps.

Docker-Compose Installation

In addition to Docker, it is required to install “docker-compose”:

In a fresh Ubuntu20 installation, typing “docker-compose” in a terminal window should provide the recommended command to install docker-compose.

(typically something like “sudo apt install docker-compose”)

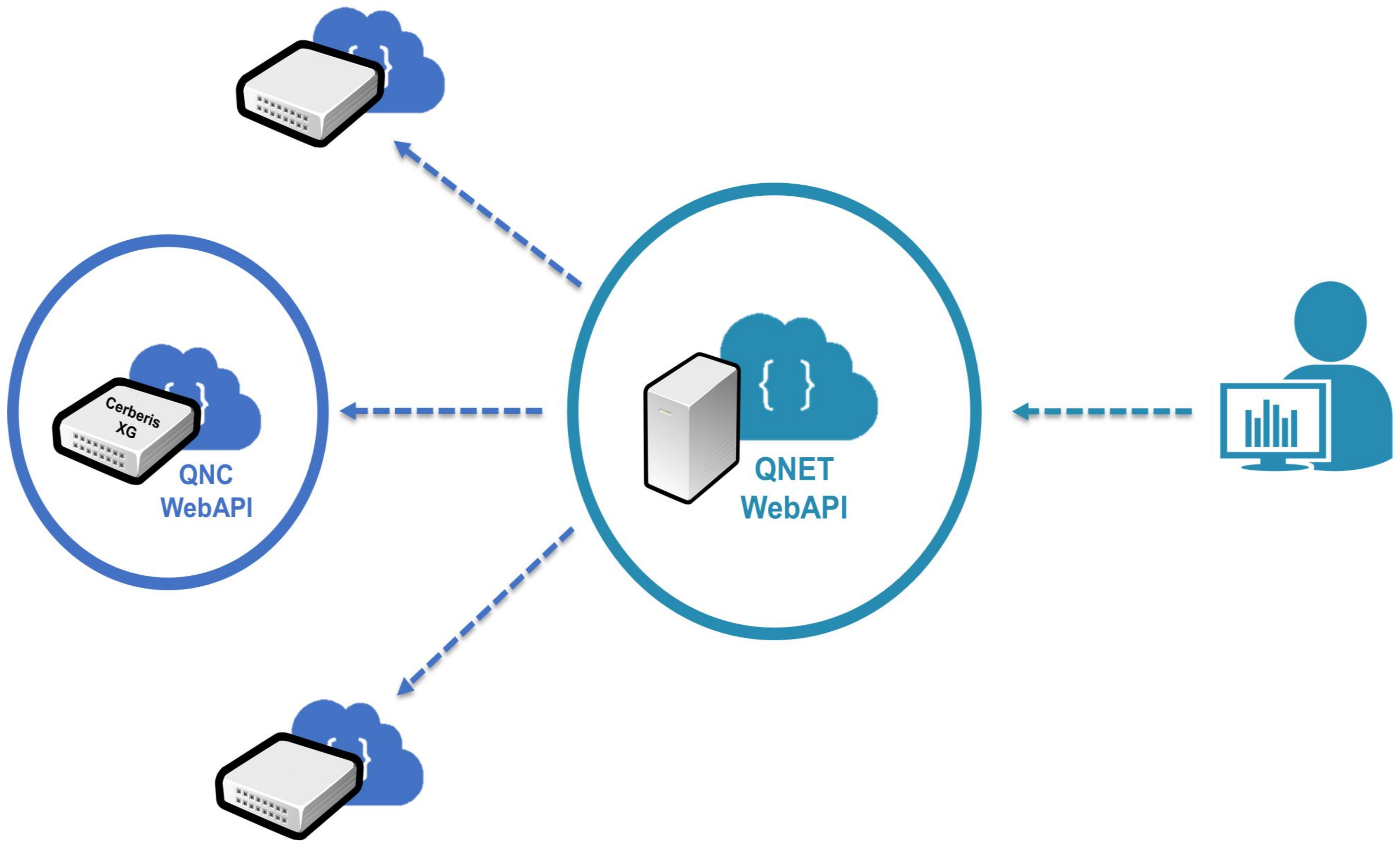
For more information, visit:

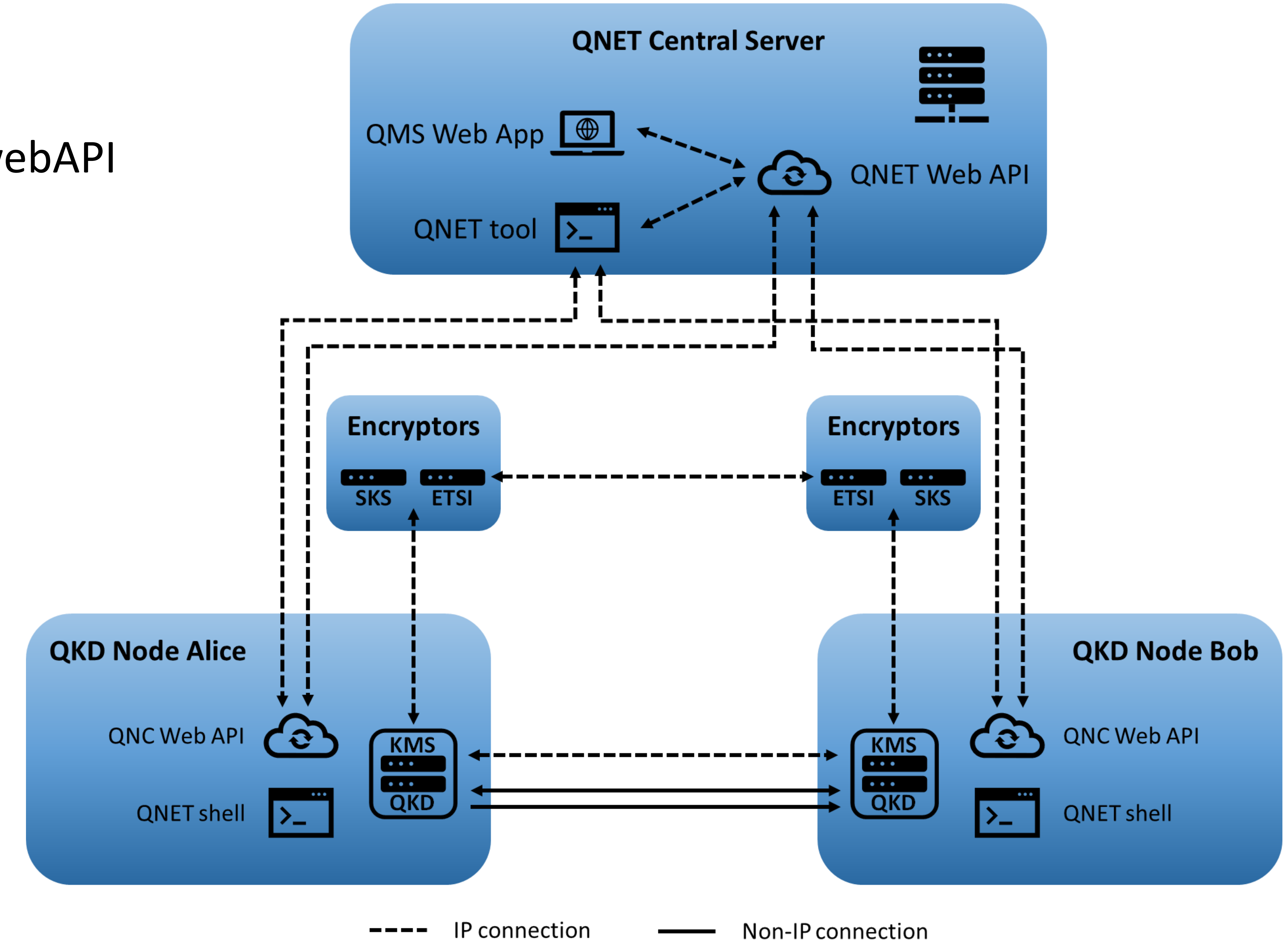
https://docs.docker.com/compose/

Installing QNET Central Server for configuring and monitoring

To monitor/configure the Cerberis XG it is necessary to install the QNET Central Server composed by:

- QNET webAPI that has two main purposes:
  - simplify QKD node's configuration and monitoring.
  - automate QKD node’s network configuration.
- QMS Web Application is a Graphical User Interface of the QNET webAPI
- QNET CLI is a command line interface of QNET webAPI





QNET/QMS Update

- To update an existing QNET/QMS installation, move to the installed QMS directory (for example /home/opt/idq)
- To update QMS/QNET from a previous version, perform the following commands:
- Verify the current docker containers are shut down:
  - **docker-compose down**
- Prune the existing docker containers:
  - **docker system prune –f**
  - **docker system prune –a**
- Remove old volumes:
  - **rm -rf volumes/qnet-* volumes/qms-***

QNET Installation

- To install QNET move the installer to the desired directory (/home/opt/idq ?)
- If necessary, add executable permissions to the file using the command:
  - ***chmod +x qnet-installer-x.x-linux-x64.run***
- Run the installer with "sudo" permissions:
  - ***sudo ./qnet-installer-x.x-linux-x64.run***

```
chris@chris-VirtualBox:/home/idq$ sudo ./qnet-installer-1.1.74.24404-20210715-linux-x64.run
Verifying archive integrity...  100%   MD5 checksums are OK. All good.
Uncompressing Installer for QNET, an administration tool for quantum networks  100%
===============================================================================
  ___  _   _ _____ _____     _           _        _ _
 / _ \| \ | | ____|_   _|   (_)_ __  ___| |_ __ _| | |
| | | |  \| |  _|   | |_____| | '_ \/ __| __/ _` | | |
| |_| | |\  | |___  | |_____| | | | \__ \ || (_| | | |
 \__\_\_| \_|_____| |_|     |_|_| |_|___/\__\__,_|_|_|

                              For qnet version 1.1.74
===============================================================================

====  Checking if current user has root privileges  ====
You have root privileges. OK.

====  Check if QNET is already installed  ====
Found binary folder [/opt/qnet] already existing.
QNET has already been installed on your system.
A new installation will erase the current one.
Start new installation of QNET.

====  Remove current QNET installation  ====
Erase folder [/opt/qnet].
Folder [/opt/qnet] successfully erased. OK.

====  Install QNET binaries  ====
Destination folder is [/opt/qnet/bin].
Update permissions.
Installed QNET binaries. OK.

====  Create script to add QNET to path  ====
Script created at [/etc/profile.d/qnet.sh]. OK.

-------------------------------------------------
QNET has been installed successfully.
If the 'qnet' command is not available, please log out and in again, or just run '. /etc/profile' in your shell.

chris@chris-VirtualBox:/home/idq$
```

QMS installation

SWISS QUANTUM

QMS installation

The QMS installation package is named qnet-**qms-<version>-<timestamp>.tar.gz**. To decompress it, use these commands in a shell:

- ***tar xvzf qnet-qms-x.y.z-ttt.tar.gz***

```
chris@chris-VirtualBox:/home/idq$ sudo tar xvf qnet-qms-1.1.5.tar.gz
[sudo] password for chris:
qnet-package-qms-1.1.5/volumes/
qnet-package-qms-1.1.5/volumes/certs/
qnet-package-qms-1.1.5/volumes/certs/IDQCustomersCA-CorpRootCA.crt
qnet-package-qms-1.1.5/volumes/certs/KMSConfigApi.pfx
qnet-package-qms-1.1.5/volumes/certs/README.txt
qnet-package-qms-1.1.5/volumes/certs/kems.crt
qnet-package-qms-1.1.5/volumes/certs/kems.key
qnet-package-qms-1.1.5/volumes/certs/qms.chained.crt
qnet-package-qms-1.1.5/volumes/certs/qms.key
qnet-package-qms-1.1.5/volumes/certs/qnet-oauth.pfx
qnet-package-qms-1.1.5/README.md
qnet-package-qms-1.1.5/docker-compose.override.yml
qnet-package-qms-1.1.5/docker-compose.yml
qnet-package-qms-1.1.5/.env
qnet-package-qms-1.1.5/images.tar
qnet-package-qms-1.1.5/volumes/certs/
qnet-package-qms-1.1.5/volumes/certs/IDQCustomersCA-CorpRootCA.crt
qnet-package-qms-1.1.5/volumes/certs/KMSConfigApi.pfx
qnet-package-qms-1.1.5/volumes/certs/README.txt
qnet-package-qms-1.1.5/volumes/certs/kems.crt
qnet-package-qms-1.1.5/volumes/certs/kems.key
qnet-package-qms-1.1.5/volumes/certs/qms.chained.crt
qnet-package-qms-1.1.5/volumes/certs/qms.key
qnet-package-qms-1.1.5/volumes/certs/qnet-oauth.pfx
qnet-package-qms-1.1.5/volumes/certs/README.txt
qnet-package-qms-1.1.5/volumes/certs/kems.crt
qnet-package-qms-1.1.5/volumes/certs/KMSConfigApi.pfx
qnet-package-qms-1.1.5/volumes/certs/IDQCustomersCA-CorpRootCA.crt
qnet-package-qms-1.1.5/volumes/certs/qnet-oauth.pfx
qnet-package-qms-1.1.5/volumes/certs/kems.key
qnet-package-qms-1.1.5/volumes/certs/qms.key
qnet-package-qms-1.1.5/volumes/certs/qms.chained.crt
chris@chris-VirtualBox:/home/idq$
```

Note you may need to use sudo permissions

IDQ

QMS Installation

Move to the newly-created directory using the command: ***cd qnet-package***

Use this command to load the images into docker:
sudo docker load -i images.tar

In the “customization” directory, the configuration *docker-compose.override.yml* provides configuration parameters. If you have special customizations to use, you can edit this file. Otherwise, for a standard installation, just leave the file as it is.

Modify the .env file in the “customization” directory to include your proper time zone: ***sudo vi ./customization/.env***

Copy the files from the customization directory to the “qnet-package” directory

cp customization/.env ./
cp customization/docker-compose.override.yml ./

QMS Installation

Launch the service with this command:
docker-compose up –d

You can stop the QMS services with this command:
docker-compose down

```
chris@chris-VirtualBox:/home/idq/qnet-package-qms-1.1.5$ sudo docker-compose up -d
Creating network "qnet-package-qms-115_default" with the default driver
Creating qnet-package-qms-115_qms-frontend_1 ... done
Creating qnet-package-qms-115_qms-postgres_1 ... done
Creating qnet-package-qms-115_qms-mongo_1    ... done
Creating qnet-package-qms-115_qnet-webapi_1  ... done
Creating qnet-package-qms-115_qnet-oauth_1   ... done
Creating qnet-package-qms-115_qnet-mariadb_1 ... done
Creating qnet-package-qms-115_qms-backend_1  ... done
chris@chris-VirtualBox:/home/idq/qnet-package-qms-1.1.5$
```

QMS installation

Docker installation

- **If necessary, you can connect to the docker for advanced troubleshooting**

#**docker exec -it 3bb995c2ca6f /bin/bash** (3bb995c2ca6f is an example, it must be changed depending on each configuration)

- **To check if the containers started**

/home/idq/QMS/QMS$ **sudo docker ps -a**

```
chris@chris-VirtualBox:/home/idq/qnet-package-qms-1.1.5$ sudo docker ps
CONTAINER ID   IMAGE                        COMMAND                  CREATED         STATUS         PORTS                                                      NAMES
dc5f2a3942dc   idq/qms-backend:1-0-2        "dotnet Ems.Platform…"   2 hours ago     Up 2 hours                                                                qnet-package-qms-115_qms-backend_1
1d0c10380aa0   idq/qms-postgres:10.17       "docker-entrypoint.s…"   2 hours ago     Up 2 hours     5432/tcp                                                   qnet-package-qms-115_qms-postgres_1
ebebf7d28fec   idq/qnet-oauth:v1.1.74       "/bin/sh -c 'echo \"T…"   2 hours ago     Up 2 hours     80/tcp, 443/tcp, 0.0.0.0:8143->8143/tcp                    qnet-package-qms-115_qnet-oauth_1
6f7ea643ed68   idq/qnet-mariadb:v1.1.74     "docker-entrypoint.s…"   2 hours ago     Up 2 hours     3306/tcp                                                   qnet-package-qms-115_qnet-mariadb_1
1f47a797d2ff   idq/qms-mongo:4.0.24         "docker-entrypoint.s…"   2 hours ago     Up 2 hours     27017/tcp                                                  qnet-package-qms-115_qms-mongo_1
0776fbb3b8e1   idq/qms-frontend:1-0-2       "/docker-entrypoint.…"   2 hours ago     Up 2 hours     0.0.0.0:6060->6060/tcp, 80/tcp, 0.0.0.0:6443->6443/tcp     qnet-package-qms-115_qms-frontend_1
7d1f924dd85a   idq/qnet-webapi:v1.1.74      "/bin/sh -c 'echo \"T…"   2 hours ago     Up 2 hours     80/tcp, 0.0.0.0:443->443/tcp                               qnet-package-qms-115_qnet-webapi_1
6be666e5a7a9   idq/kems_database:v2-1-35    "docker-entrypoint.s…"   3 weeks ago     Up 2 hours     3306/tcp                                                   kems-config-api-v2-1-35_kems-database_1
chris@chris-VirtualBox:/home/idq/qnet-package-qms-1.1.5$
```

- **To check if the containers started**

/home/idq/QMS/QMS$ **sudo docker logs 1cfdd2bd6ff6**

QMS installation

Docker installation

- **If necessary (no access to web page), you can:**

- **disable SE Linux**

```
# vi /etc/selinux/config
...
SELINUX=disabled
...
```

- **disable the firewall**

```
# systemctl stop firewalld
# systemctl disable firewalld
```

Sample QNet Commands (Script):

```
#!/bin/bash

# Authenticate First

qnet config qnetwebapi -url https://localhost:443 -user Admin -pwd Admin


echo "Creating new group.... (Deleting Group if it Already Exists)"

qnet delete group Group1

qnet create group Group1

echo "Creating node....."


qnet create node CerberisXGA Group1 -type Mgmt -ip 10.10.10.102 -uid cfa8658d9b[illegible]d550c5a7360bb1

qnet create node-network-intferface CerberisXGA Key 192.168.10.1

qnet create node-network-intferface CerberisXGA Kms 192.168.30.1


qnet create node CerberisXGB Group1 -type Mgmt -ip 10.10.10.107 -uid 0a210554d2c[illegible]9e8ce31da1f70

qnet create node-network-intferface CerberisXGB Key 192.168.10.2

qnet create node-network-intferface CerberisXGB Kms 192.168.30.2


qnet create node-link KMSLINK CerberisXGA CerberisXGB


qnet create provider QKDA CerberisXGA Cerberis3 -type any -ip 192.168.20.1

qnet create provider QKDB CerberisXGB Cerberis3 –type any -ip 192.168.20.2


qnet create provider-link PROVLINK QKDA QKDB --key-req-interval 80


qnet create consumer CiscoA CerberisXGA CISCO --key-type QKEY --pport 9999

qnet create consumer CiscoB CerberisXGB CISCO --key-type QKEY --pport 9999


qnet create consumer ETSIA CerberisXGA Etsi --key-type QKEY --subject-dn "C=CH,ST=Geneva,L=Geneva,O=ID Quantique,OU=QuantumSafe,CN=QKDServer" --pport 443

qnet create consumer ETSIB CerberisXGB Etsi --key-type QKEY --subject-dn "C=CH,ST=Geneva,L=Geneva,O=ID Quantique,OU=QuantumSafe,CN=QKDServer" --pport 443


qnet create consumer-link CiscoLINK CiscoA CiscoB --psk-ascii Cisco1234

qnet create consumer-link ETSILINK ETSIA ETSIB --default-key-size 256


qnet create path CiscoA CiscoB -mode Automatic

qnet create path ETSIA ETSIB -mode Automatic


qnet add ca-cert consumer --ca-file ChrisCA.pem

qnet add cert QKDServer.pem QKDServer_pkey.pem


qnet deploy group Group1


qnet update consumer ETSIA --cert-file QKDServer.pem --key-file QKDServer_pkey.pem

qnet update consumer ETSIB --cert-file QKDServer.pem --key-file QKDServer_pkey.pem


qnet deploy group Group1
```

Before Moving Forward with QMS Configuration, It may be Quicker/Easier to use a QNET Configuration Script Instead:

The following slides will detail both options.

(Note you will need to configure Node name in CLI using “qnc --set-name” command)

Note that Clavis3 uses provider type “Cerberis3” and <u>not</u> “Clavis300”.

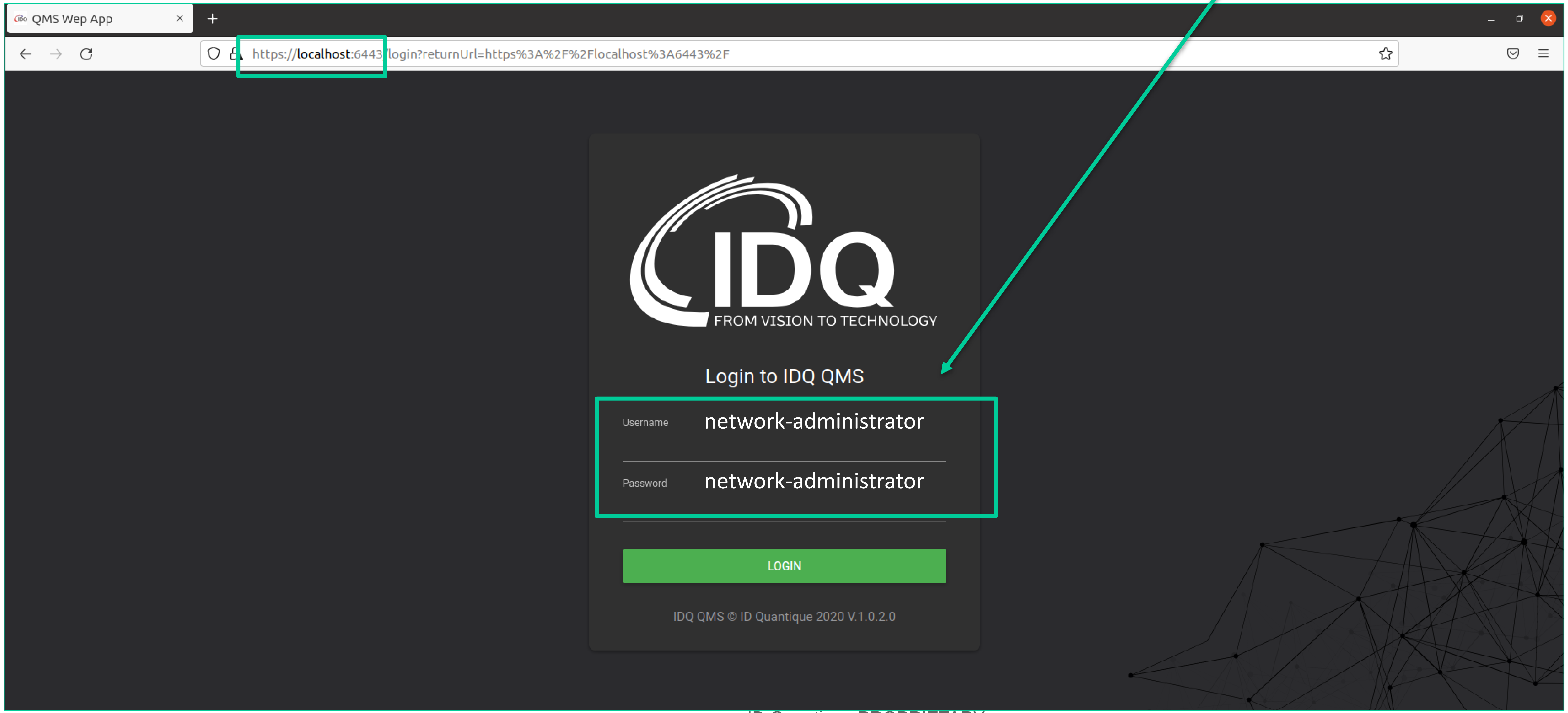
QMS configuration

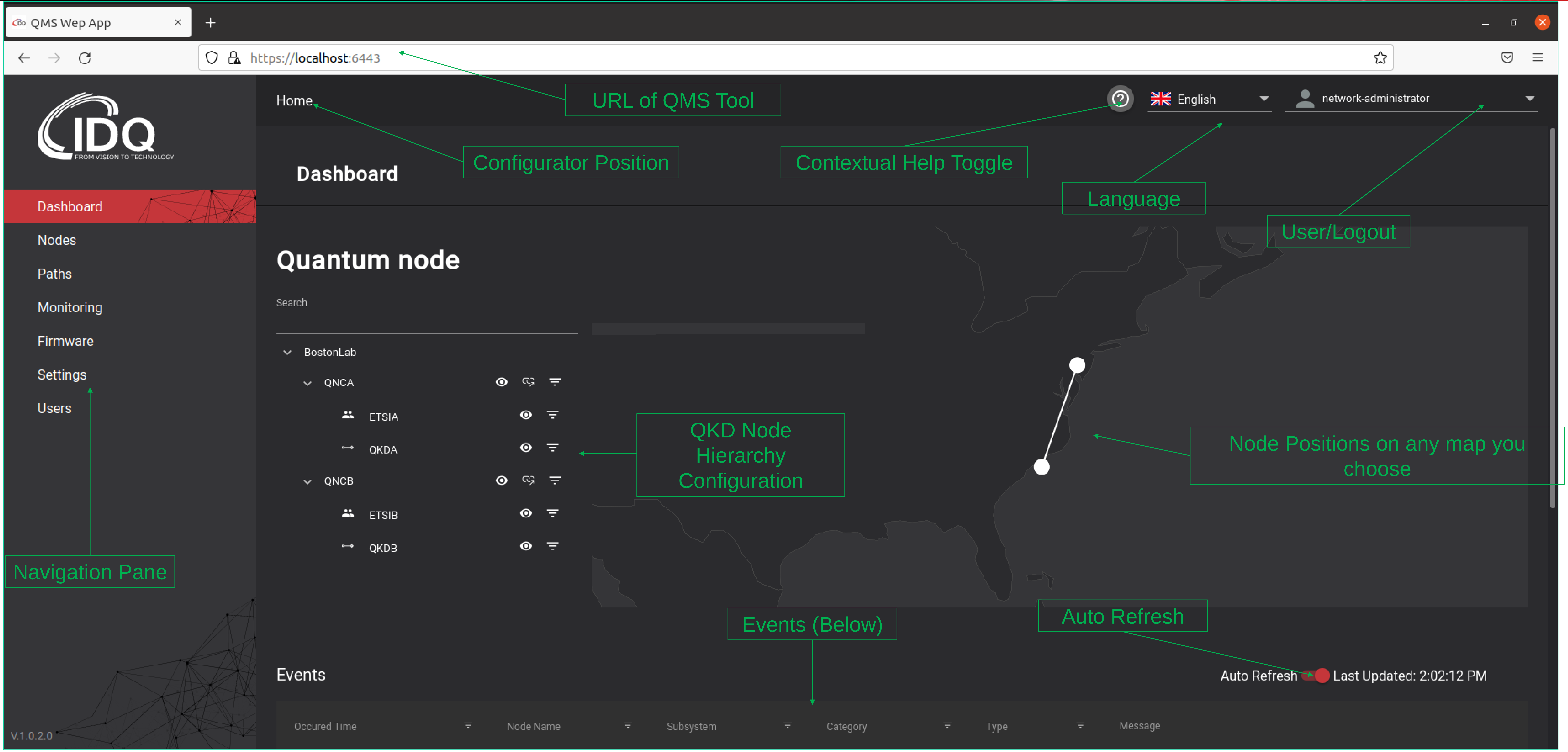
QMS installation

Access to the GUI

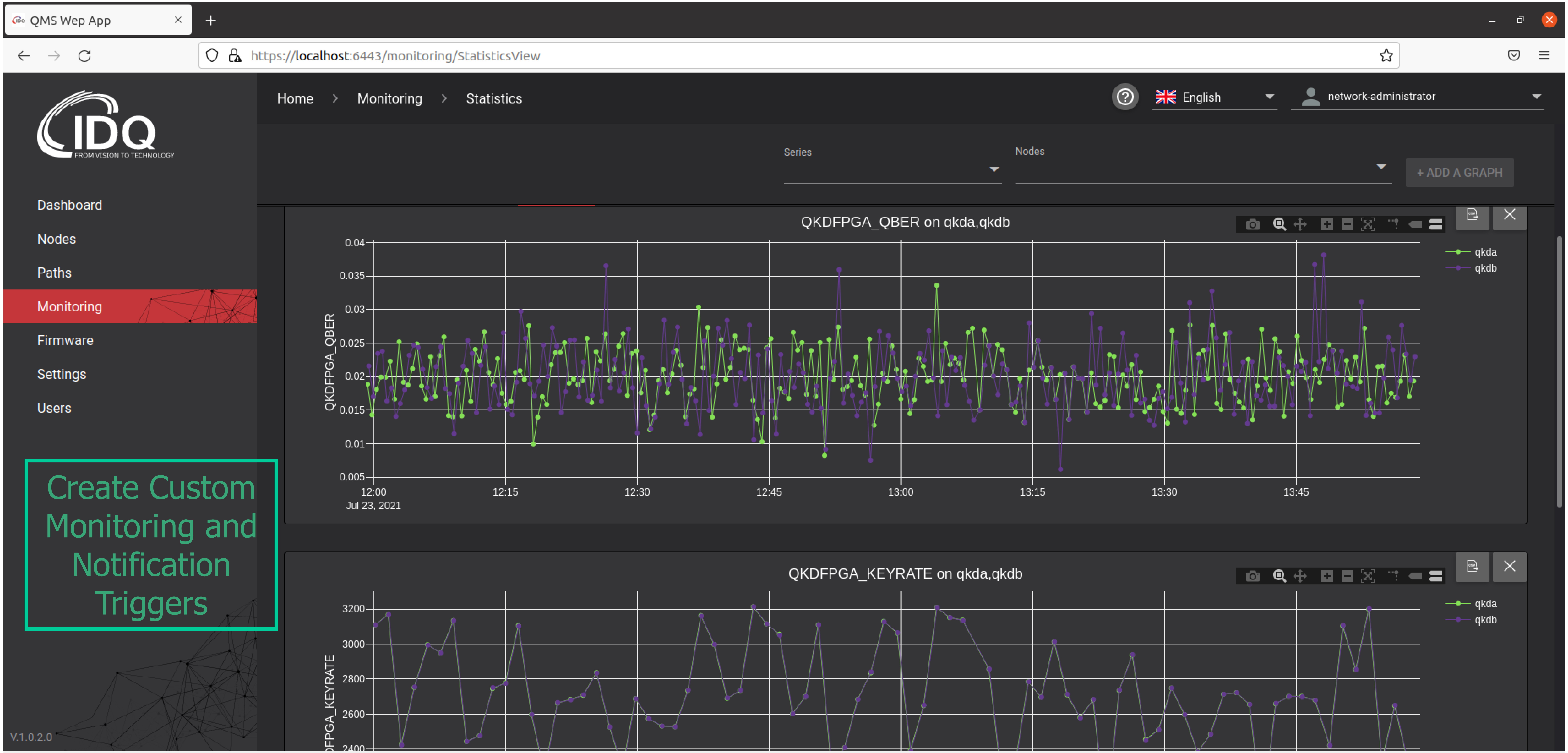
Need to type "https"

- **You should be able to access the webpage** (https://x.x.x.x:6443)
- **Default Credentials: network-administrator/network-administrator**




QMS Wep App
https://localhost:6443
URL of QMS Tool
Home
Configurator Position
Contextual Help Toggle
English
Language
network-administrator
User/Logout
Dashboard
Nodes
Paths
Monitoring
Firmware
Settings
Users
Navigation Pane
Quantum node
Search
BostonLab
QNCA
ETSIA
QKDA
QNCB
ETSIB
QKDB
QKD Node Hierarchy Configuration
Node Positions on any map you choose
Events (Below)
Auto Refresh
Events
Auto Refresh
Last Updated: 2:02:12 PM
Occured Time
Node Name
Subsystem
Category
Type
Message
V.1.0.2.0

Monitoring



Create Custom Monitoring and Notification Triggers

Known Limitations

- In the QMS GUI configuration, The Consumer, Provider, KMS and Links names are limited to 16 characters maximum.
- Avoid special characters such as "-" "_"

QMS installation

Naming on QMS

- **The naming is different than those seen before**

  - **KMS** → "CerberisXG" or "Cerbeis3 QNC" Acting as KMS for this site
  - **Provider** → "CerberisXG" or "Cerberis3 QKD"
  - **Consumer** → Encryptor
  - **KMS link** → Link between "CerberisXG" KMS or "Cerberis3 QNC" KMS
  - **Provider link** → Link between "CerberisXG" or "Cerberis3 QKD"
  - **Consumer link** → Link between encryptors

QMS processes

SWISS QUANTUM

Preliminary configuration

- **Set the QNC name on each CerberisXG or Cerberis3 QNC and copy the UID**

QMS Configuration

- **Group**
  - **Create QMS Group**

- **Node**
  - **Create a Node for each QKD device**
  - **Create a Node link for each QKD pair**

- **Provider**
  - **Create a Provider for each QKD device**
  - **Create a Provider link for each QDK pair**

QMS processes

- Consumer
  - Create a Consumer for each encryption device
- Path
  - Create a Consumer link and key routing path for each encryption pair
- Deploy
  - Deploy the newly-created configuration to the QKD devices
- Certificates
  - Load the QKD server certificates, keys, and CA to the CerberisXG/Cerberis3
  - Assign the certificates/keys to each consumer on CerberisXG/Cerberis3
  - Add the client certificates, keys, CA to each encryption device
- Verify
  - Check if the system is functional and exchanging keys

Prepare Node

Prepare Node/KMS

- Connect via ssh to the QNC CLI: ssh admin@x.x.x.x
- Execute the following command (set the name you want):

  qnc --set-name <system_name>

```
admin>
admin> qnc --set-name QNCA
Name QNCA set as local name
admin>
```

- Execute the following command to check the name of the KMS:

  qnc

```
admin> qnc
Product: QNC
QNC Name: QNCA
QNC UID: 12370e90d2114f42bdc669bf92b2f50c
Firmware: 0.7.0-18761-QNC
Serial Number: 35180626
admin>
```

- Write down the **UID**

Repeat the same process to prepare the peer node

Prepare Node/KMS

- Connect via ssh to the peer QNC CLI: ssh admin@x.x.x.x
- Execute the following command (set the name you want):

  qnc --set-name <system_name>

```
admin> qnc --set-name QNCB
QNCB successfully renamed
```

- Execute the following command to check the name of the KMS:

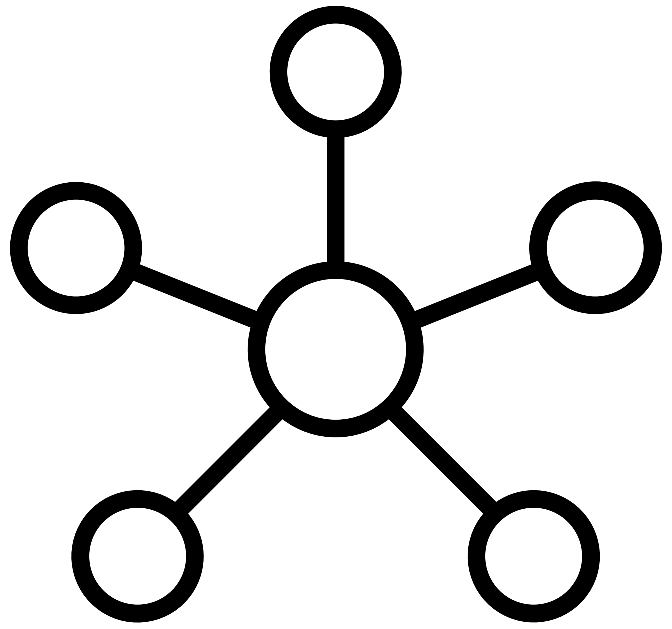
  qnc

```
admin> qnc
Product: QNC
QNC Name: QNCB
QNC UID: 0a210554d2c84dfdbfd9e8ce31da1f70
Firmware: 0.9.0-24182-QNC
Serial Number: 42180051
admin>
```

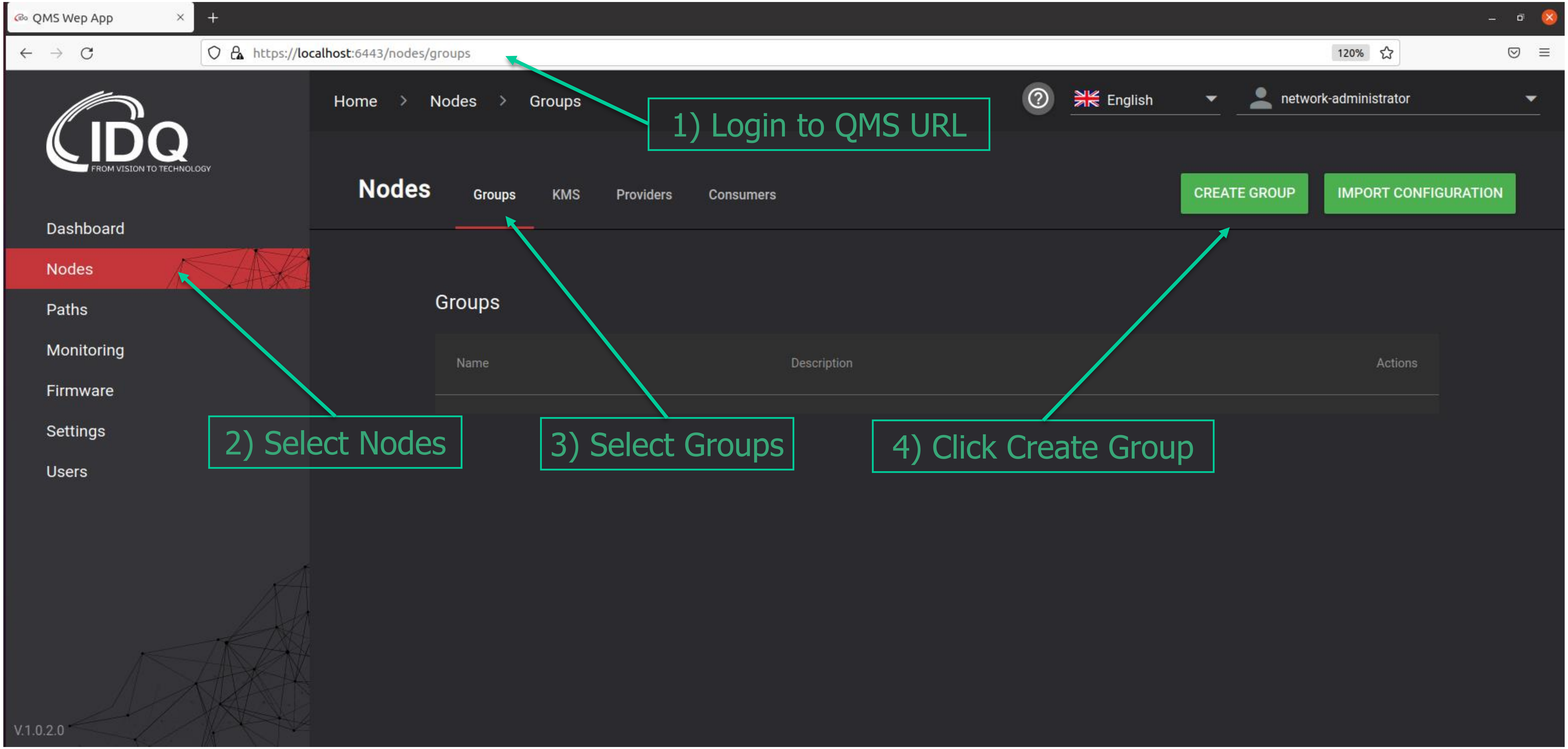
- Write down the **UID**

Create Group

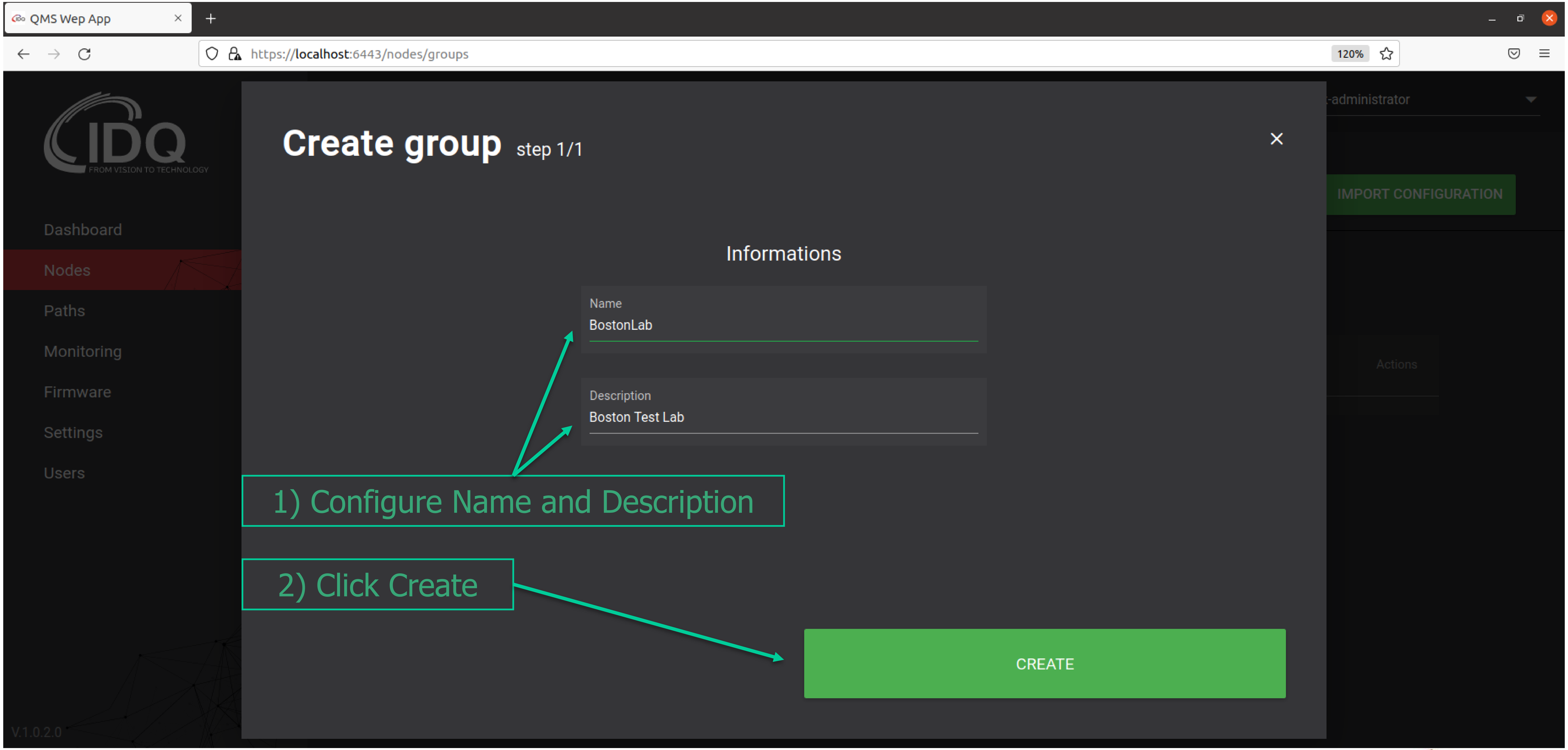
Create Group



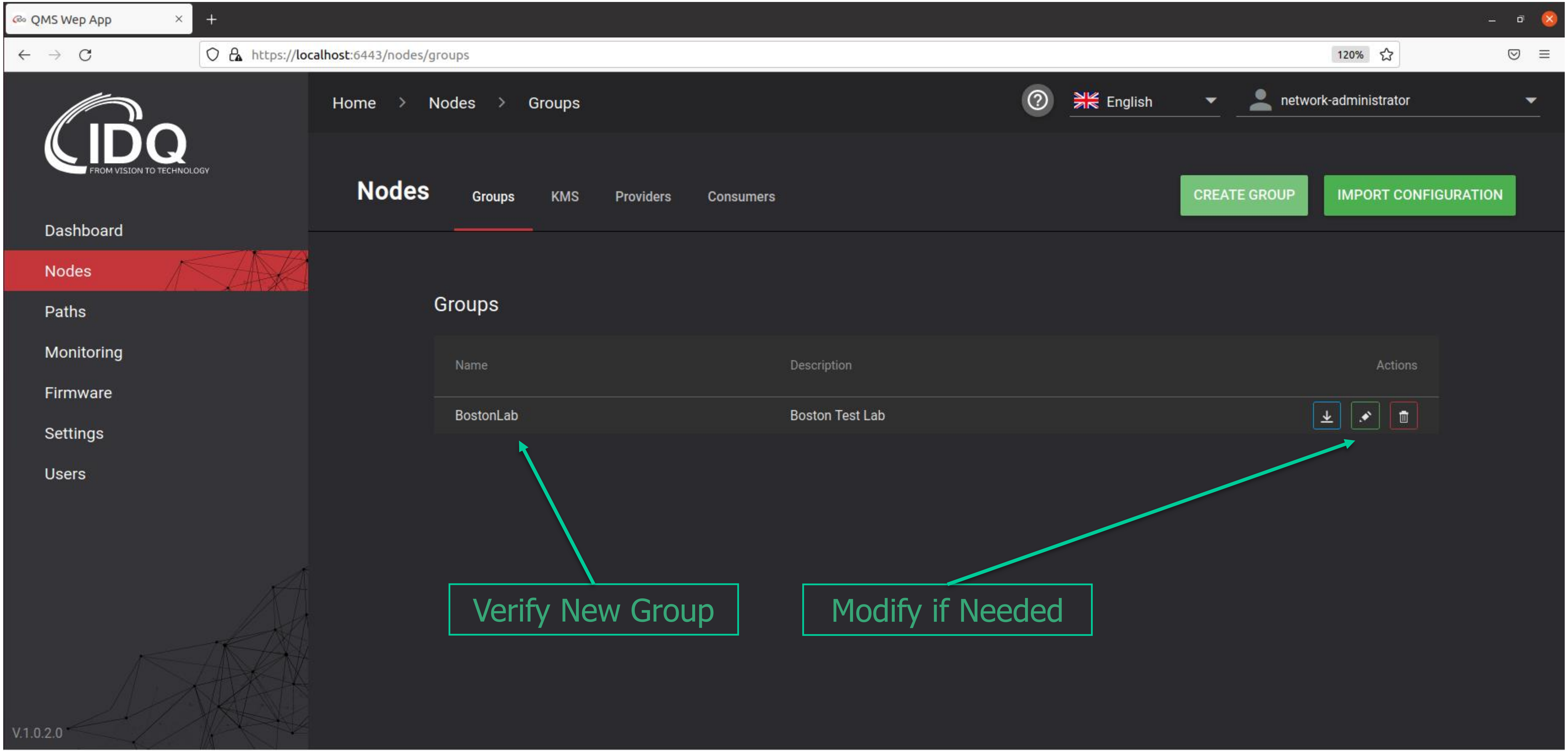
Create Group



Create Group

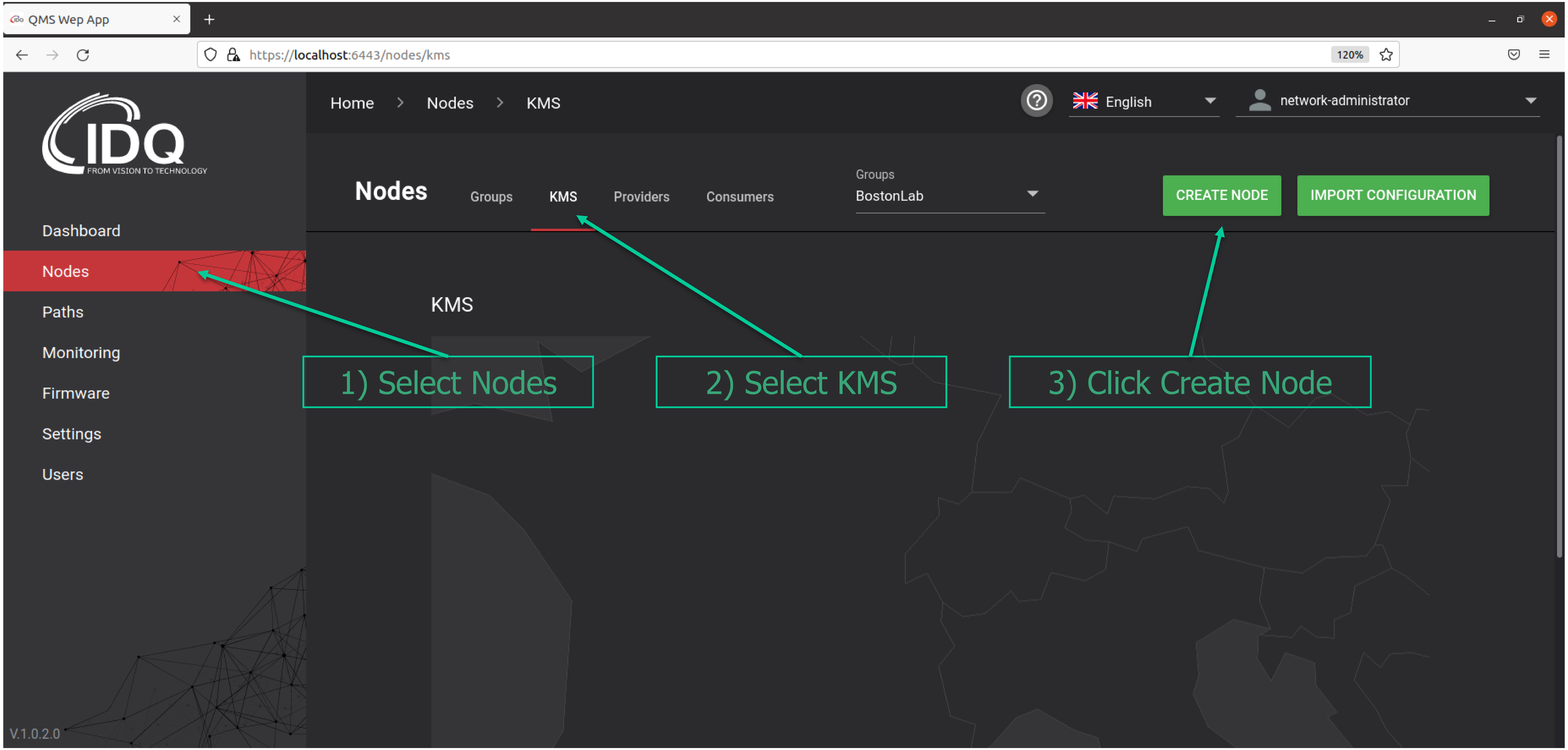


Create Group

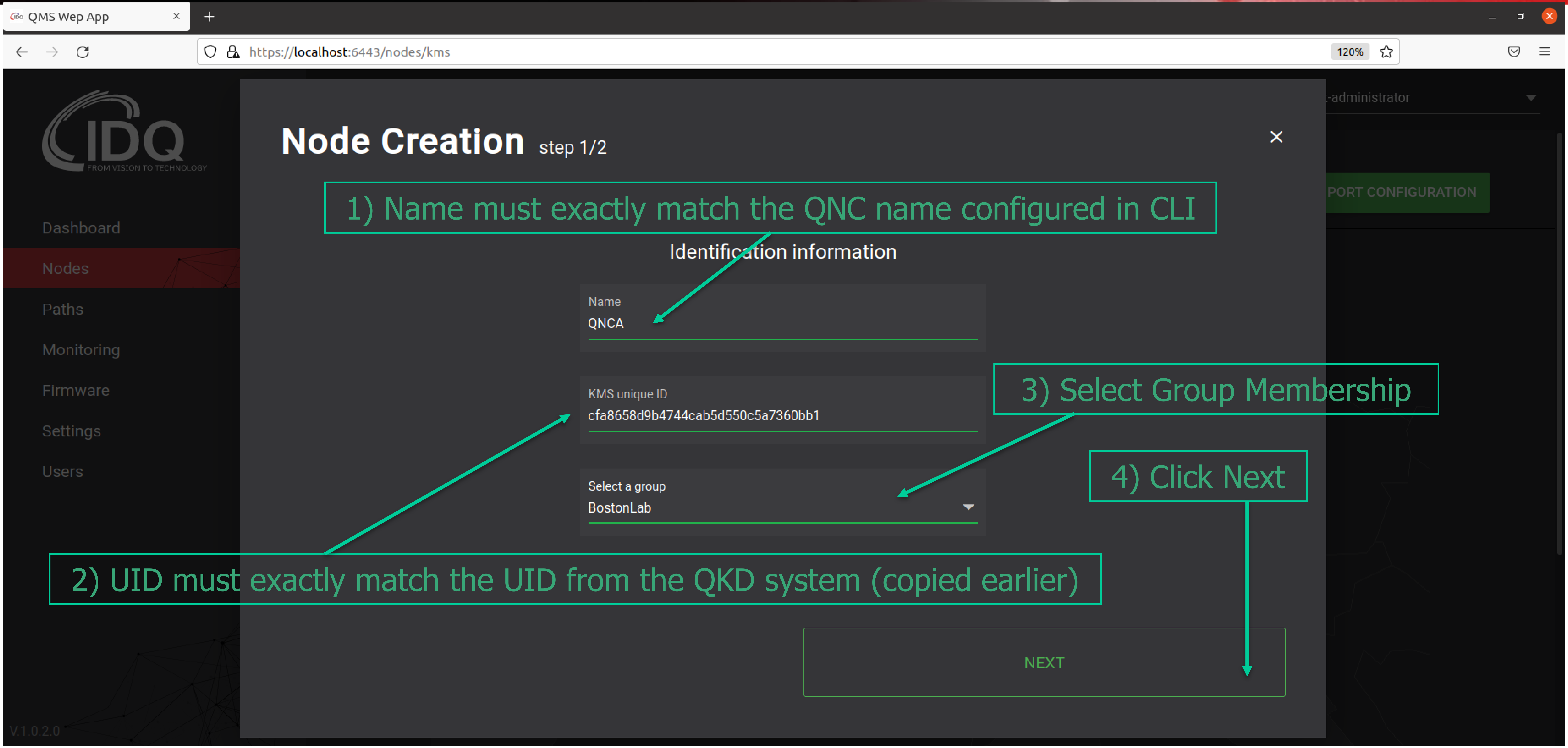


Create Nodes

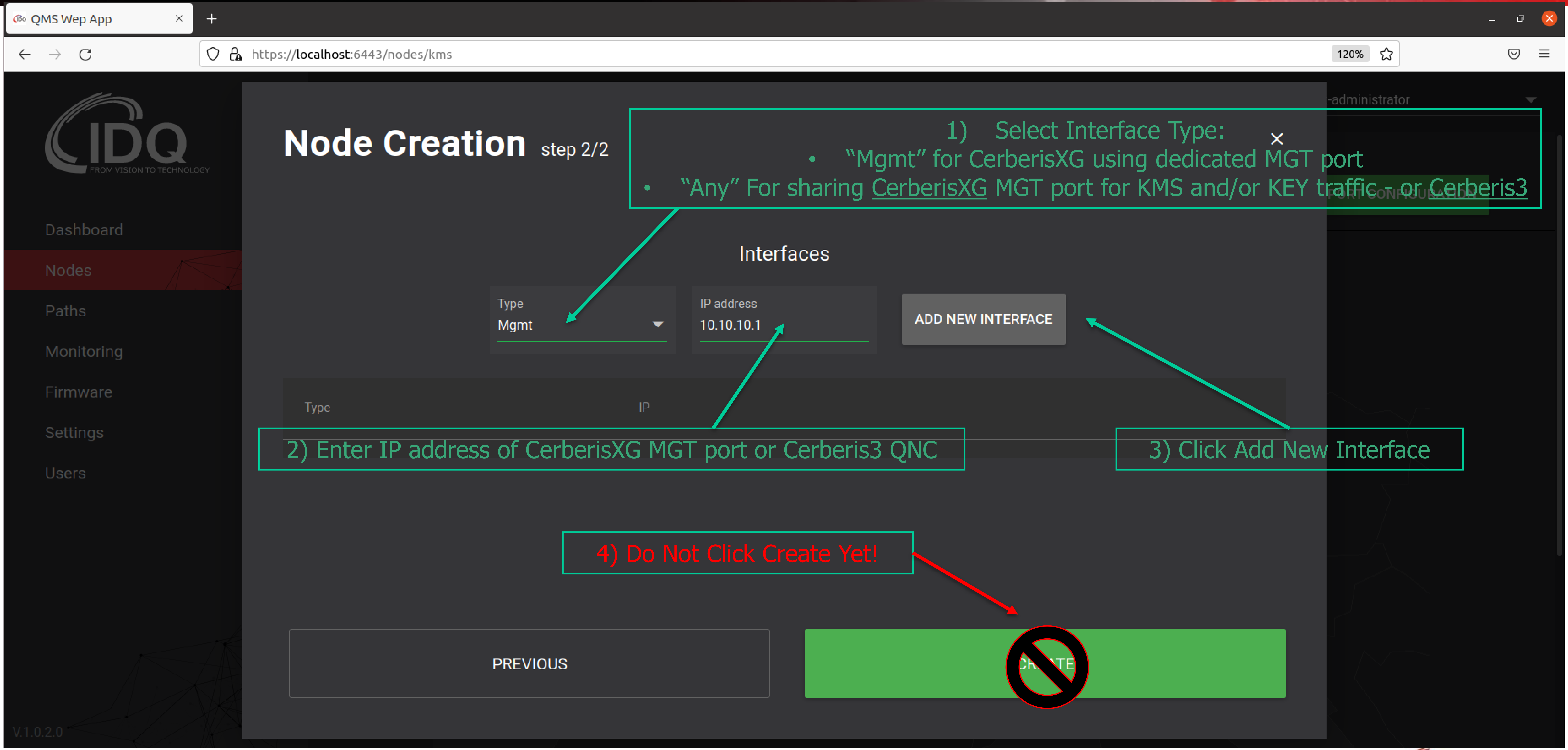
Create Nodes/KMS



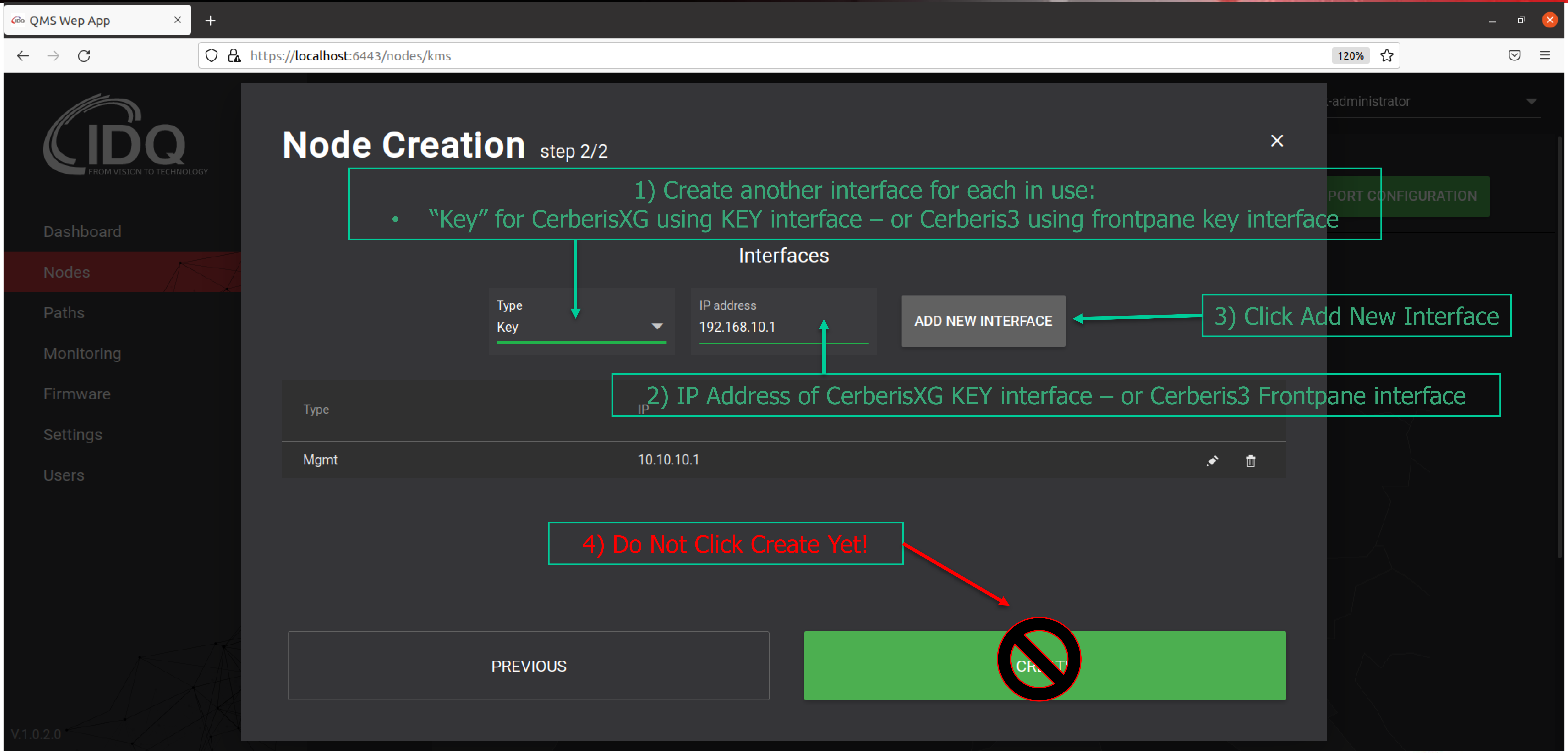
Create Nodes/KMS



Create Nodes/KMS

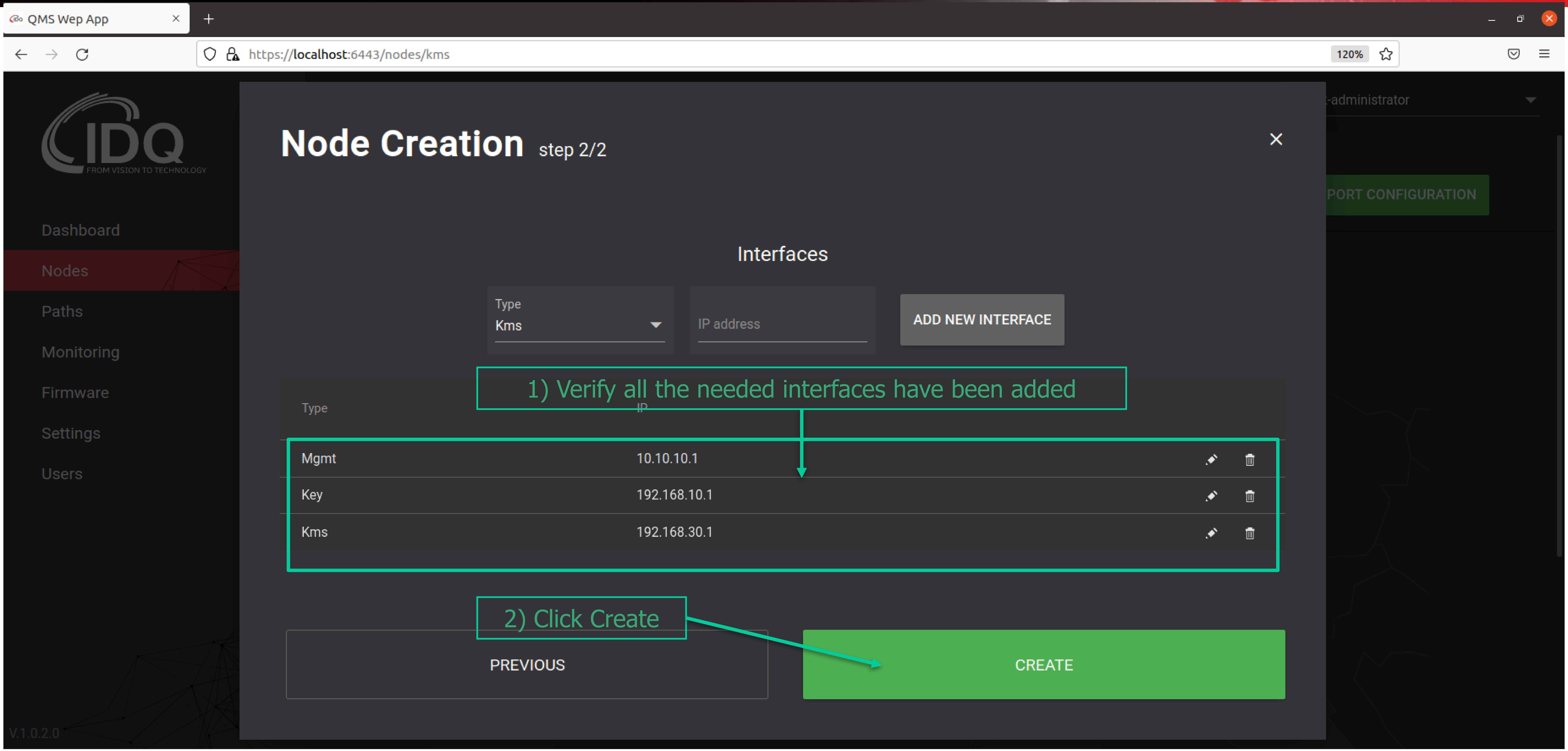


Create Nodes/KMS

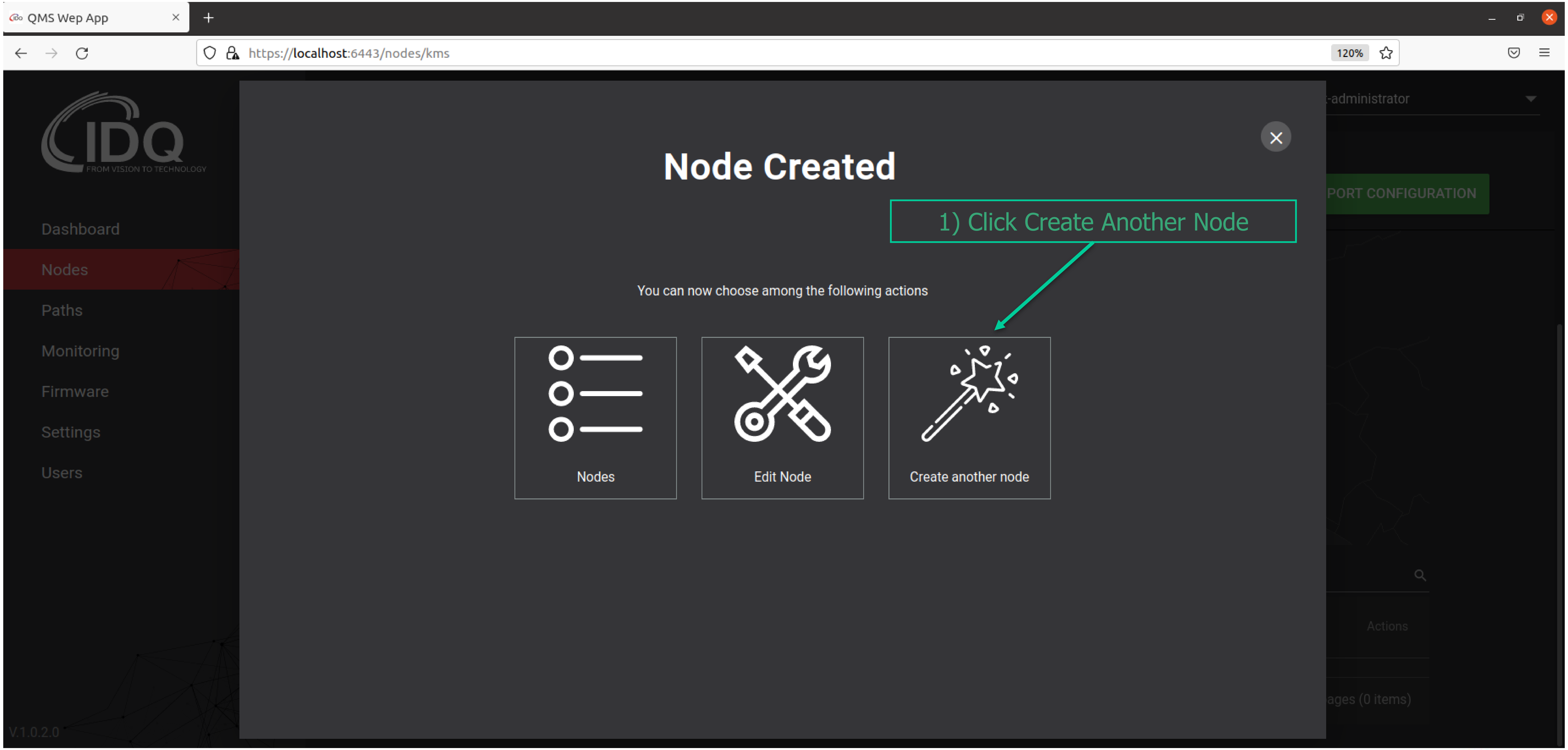


Create Nodes/KMS

Create Nodes/KMS



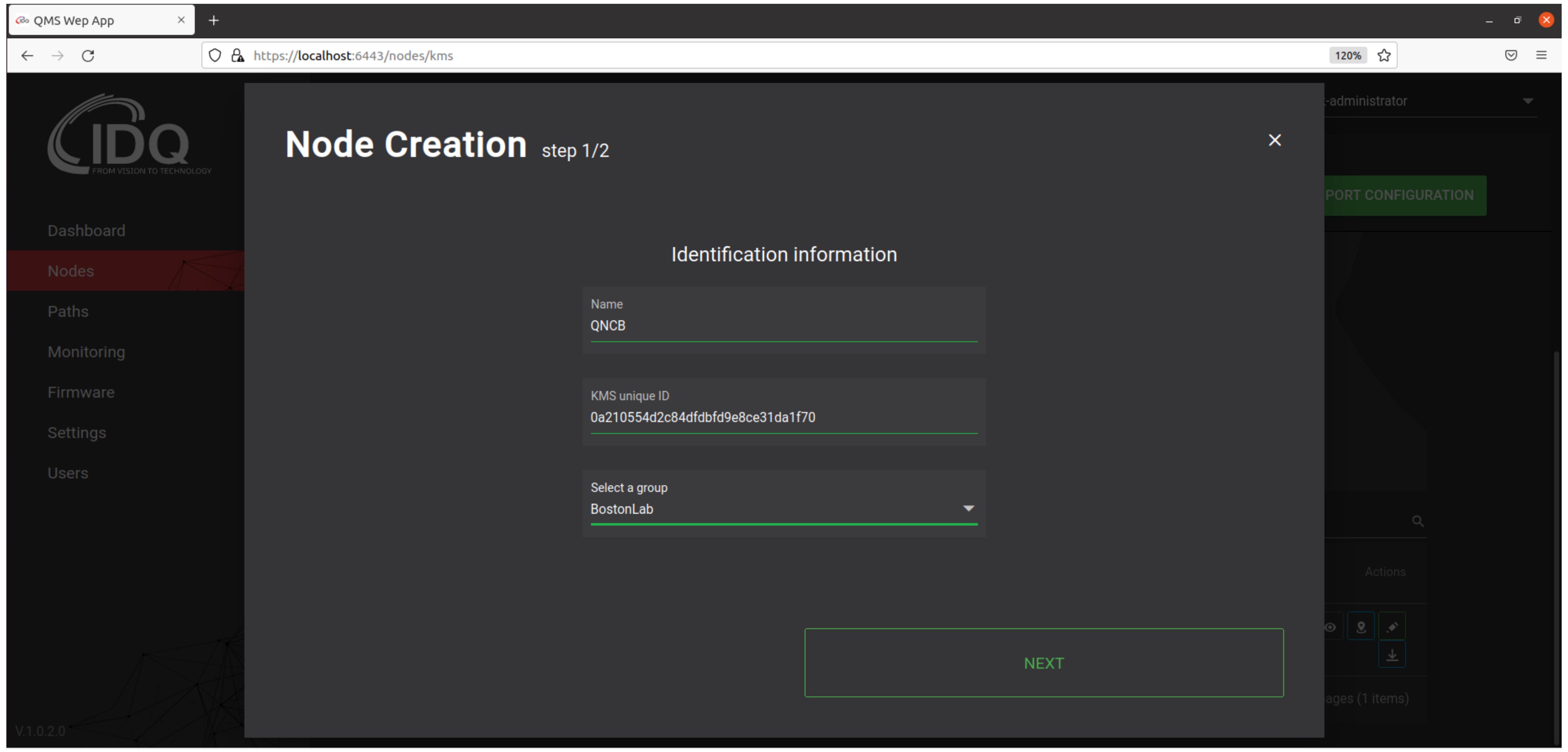
Create Nodes/KMS



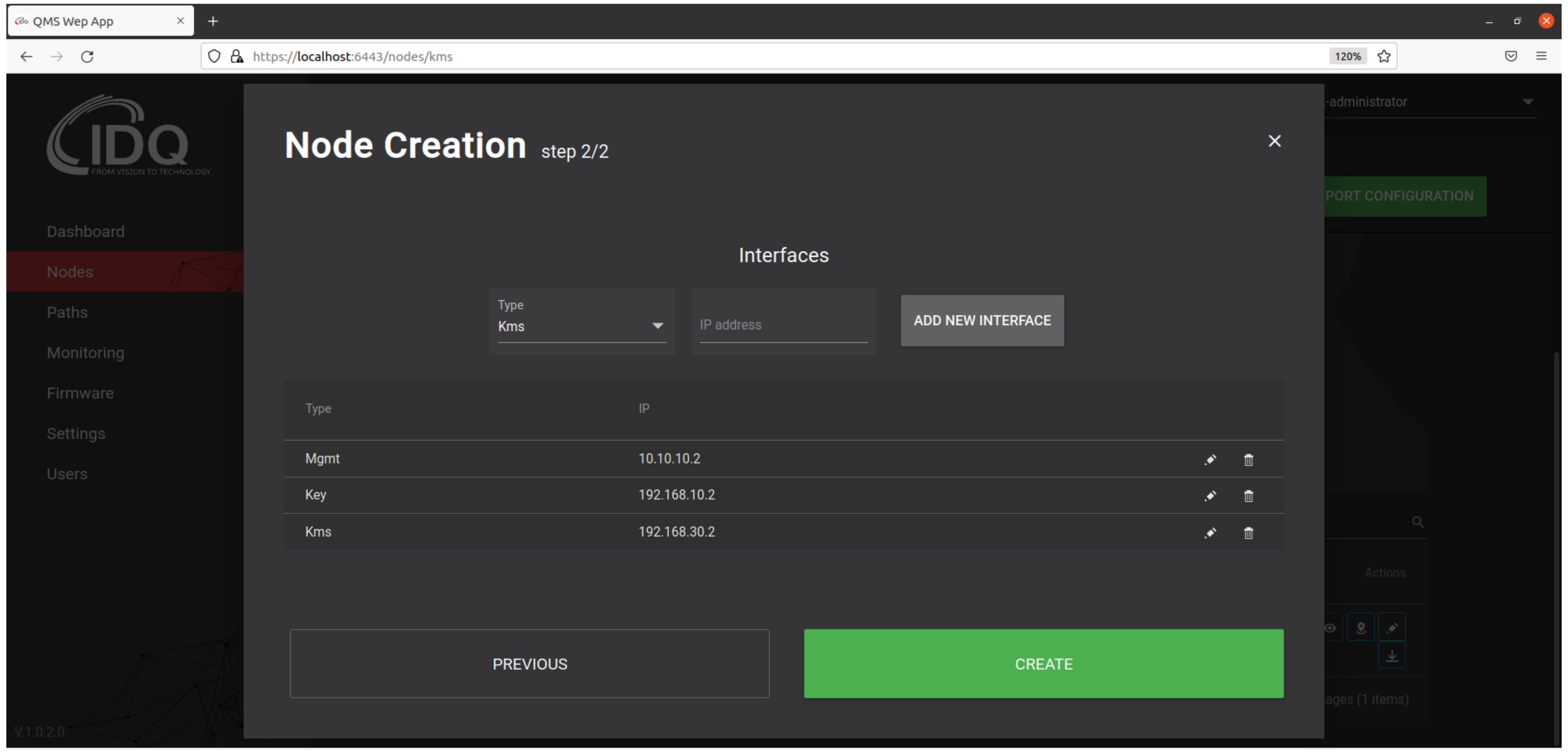
Create Nodes/KMS

Repeat the same process to create the peer node

Create Nodes/KMS



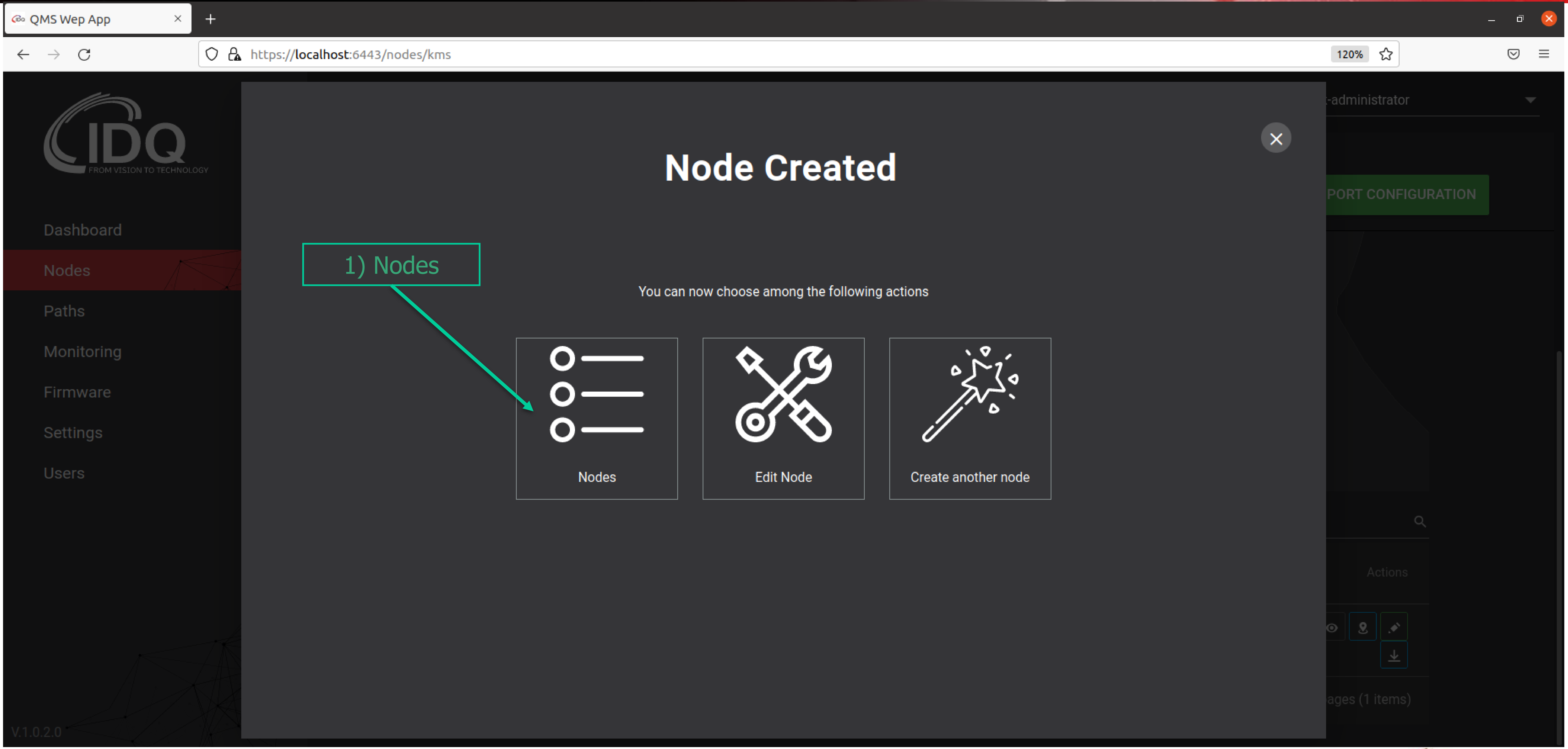
Create Peer Node.
Remember the Name and UID must exactly match the name and UID in the CLI "qnc" command



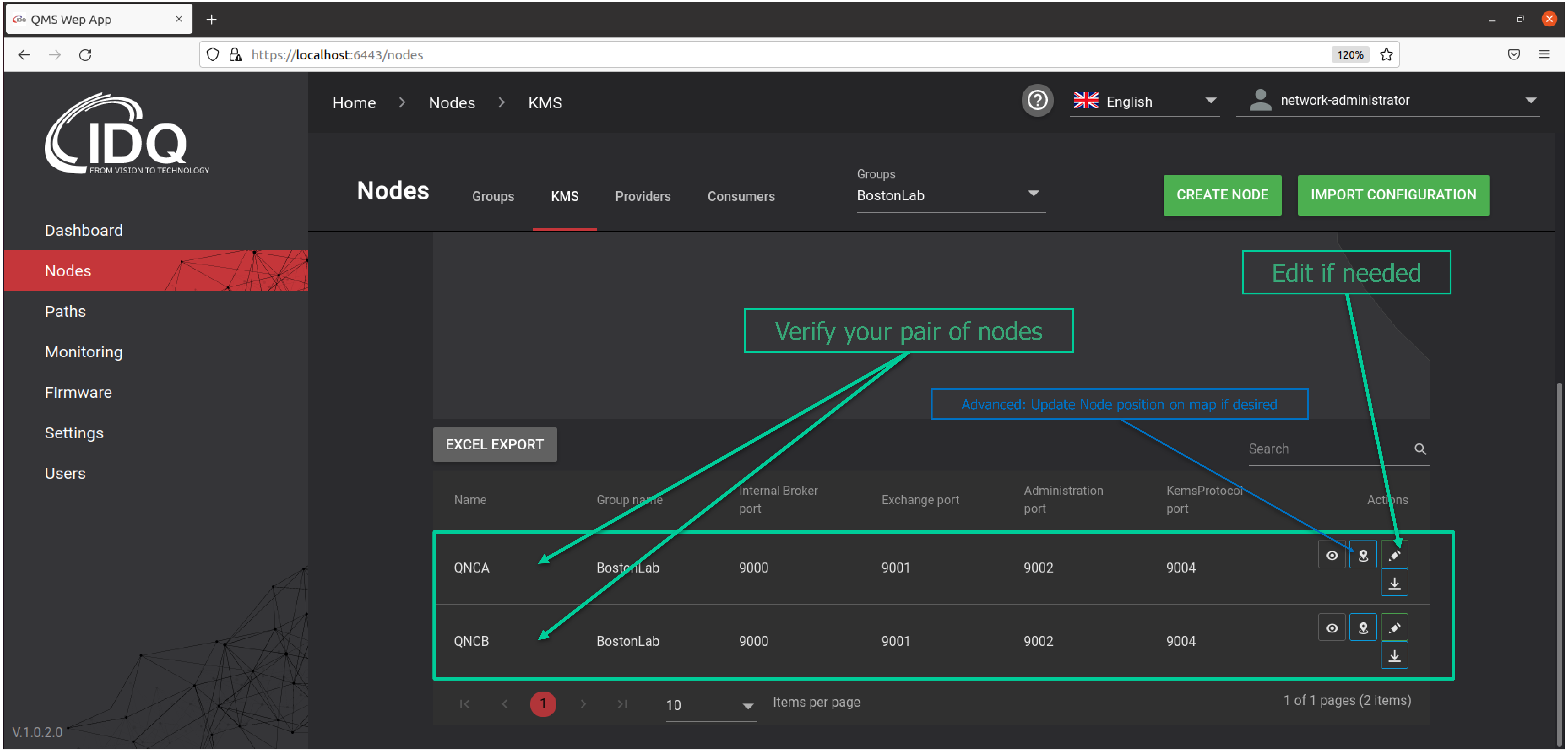
Create the necessary Interfaces
Remember: If you are sharing the same interface for MGT, KEY, KMS, you only need to create an "Any" interface.

"Any" interface is mandatory on Cerberis3 for the base management IP (Do not use type "mgmt.").

Create Nodes/KMS



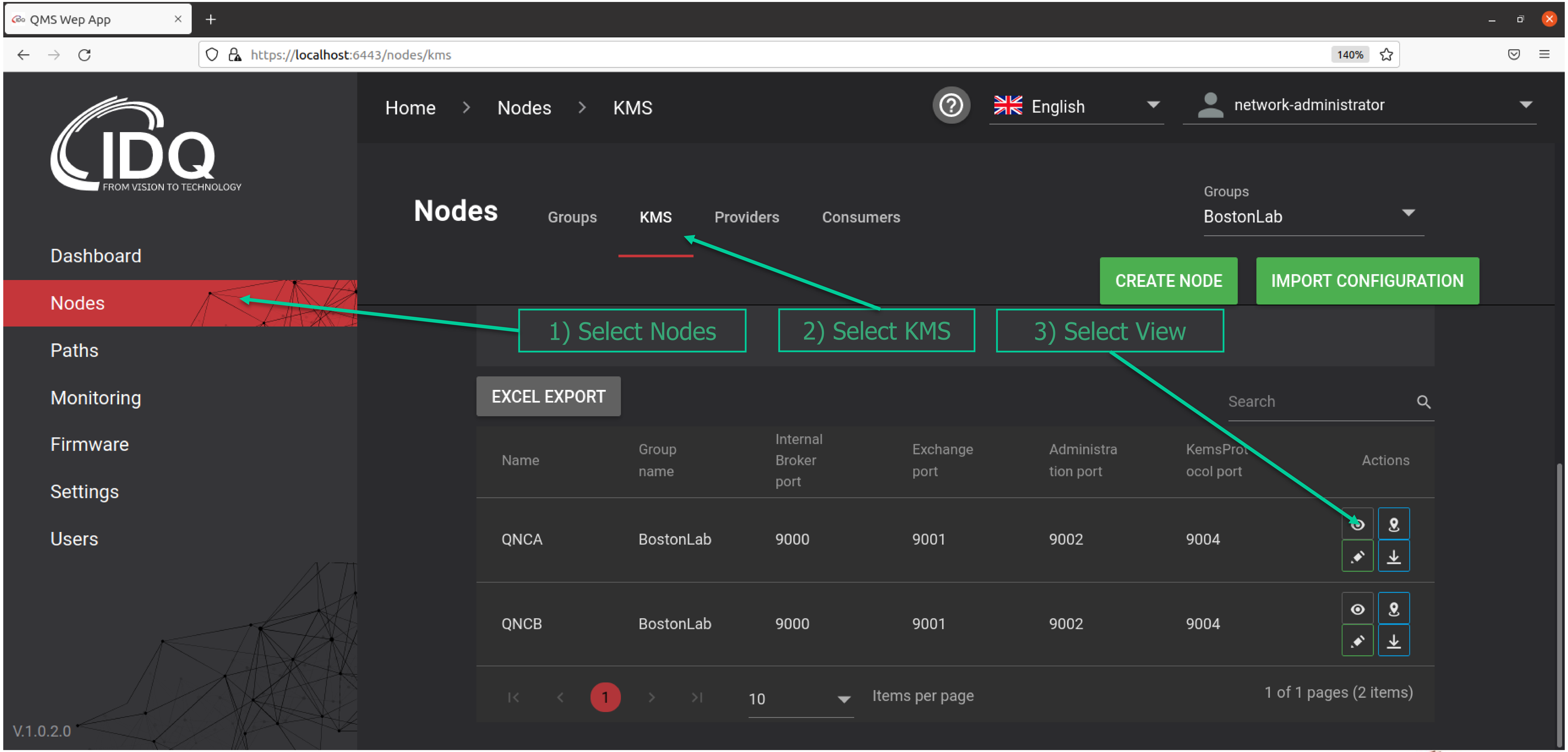
Create Nodes/KMS



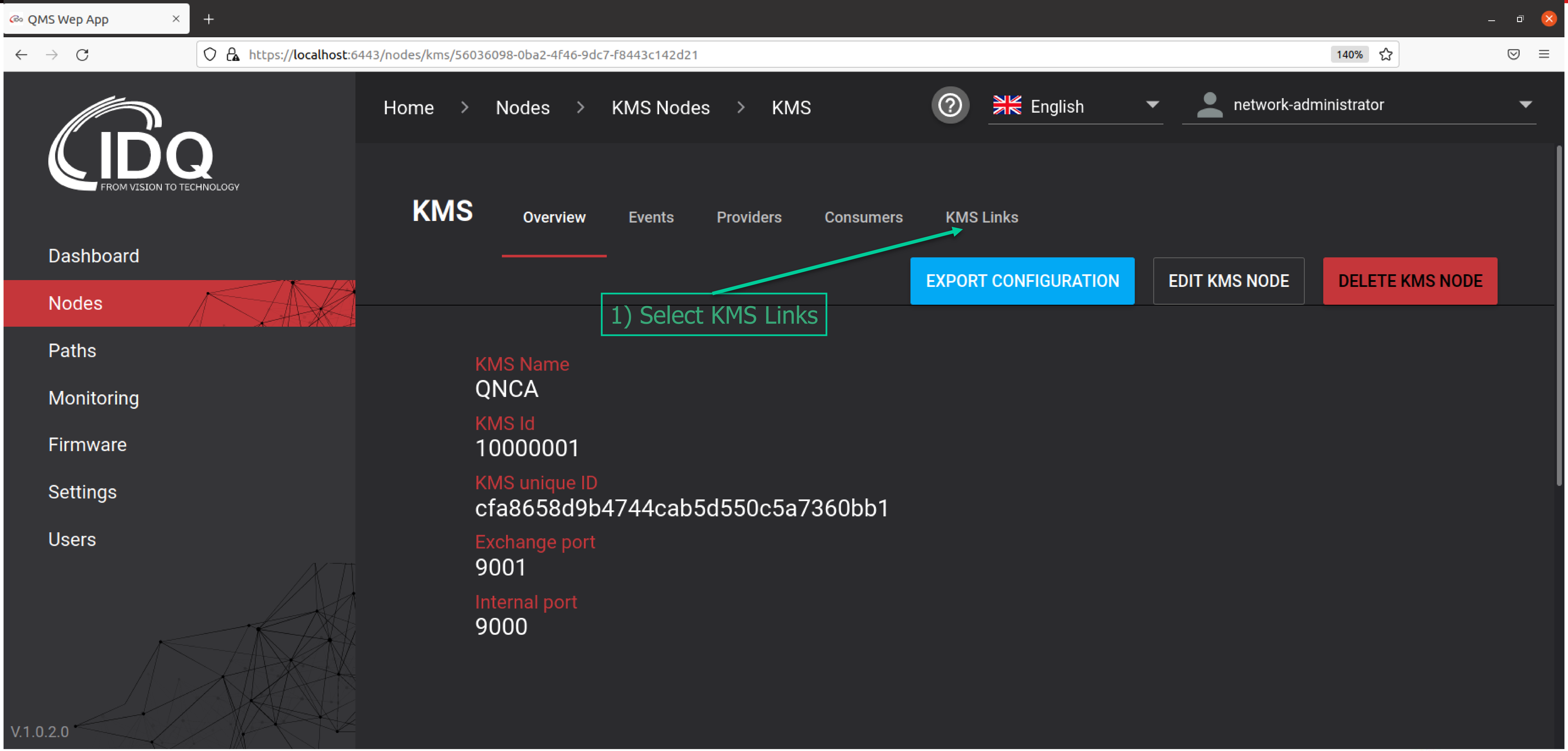
Create Node Links

Create Node KMS Links

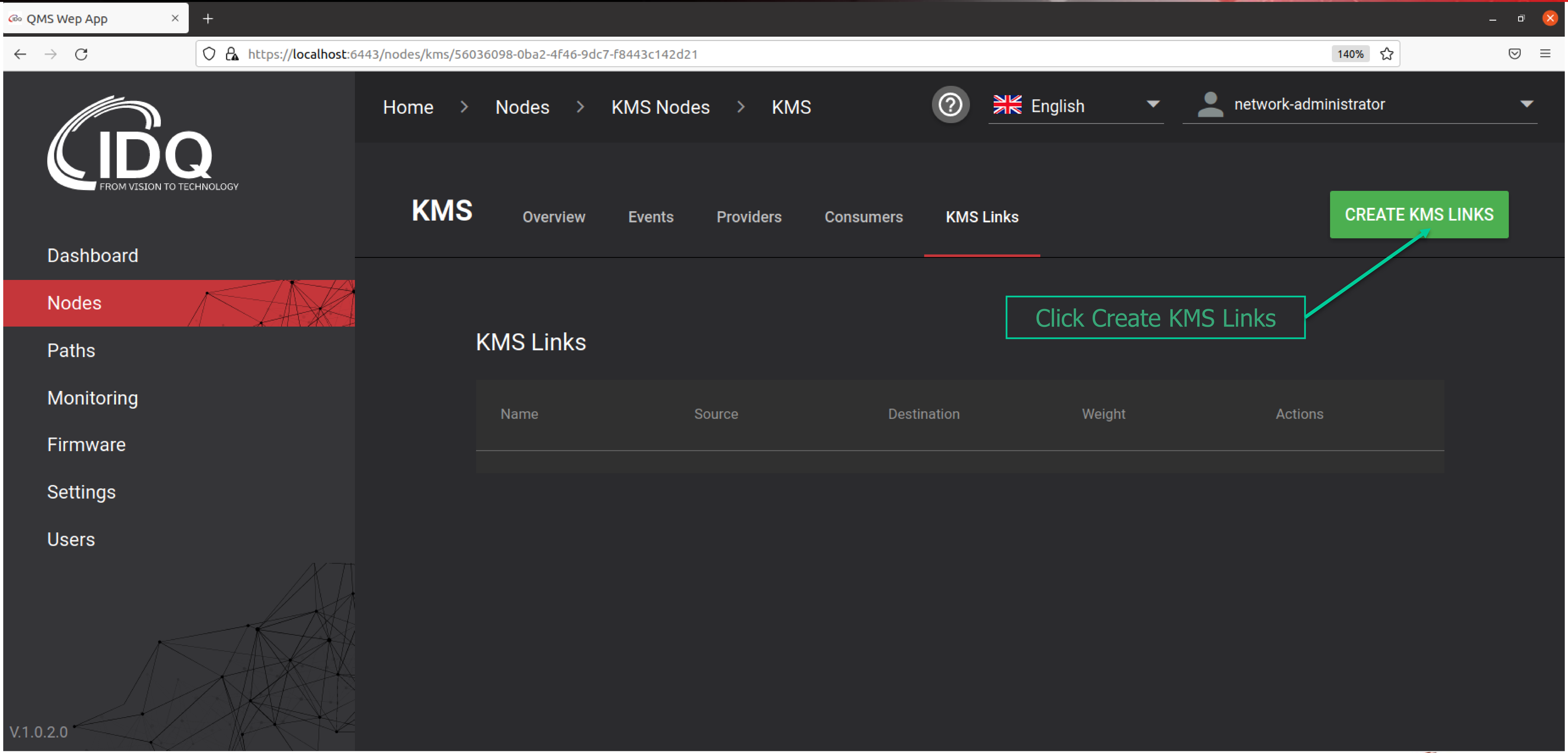
Create Node Links



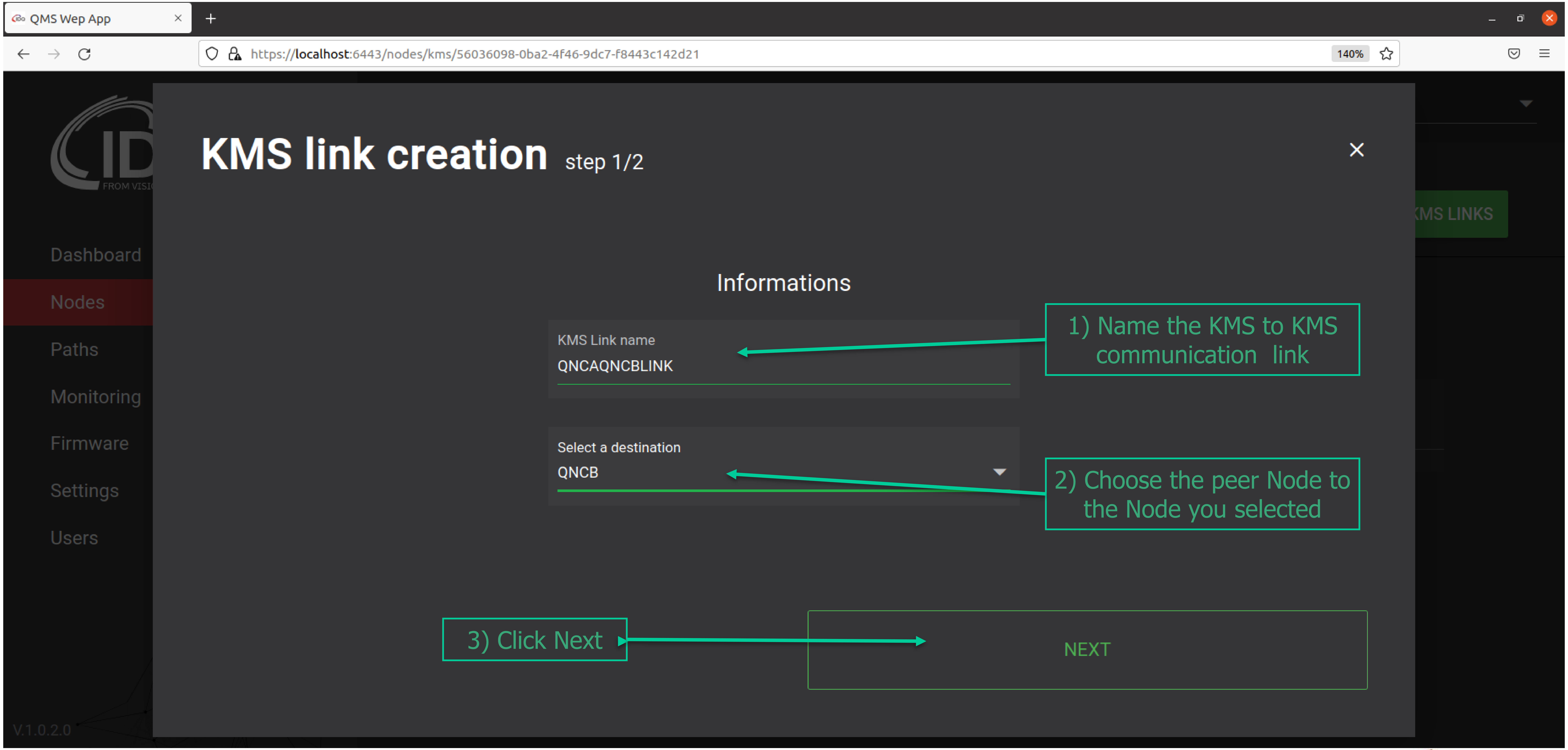
Create Node Links



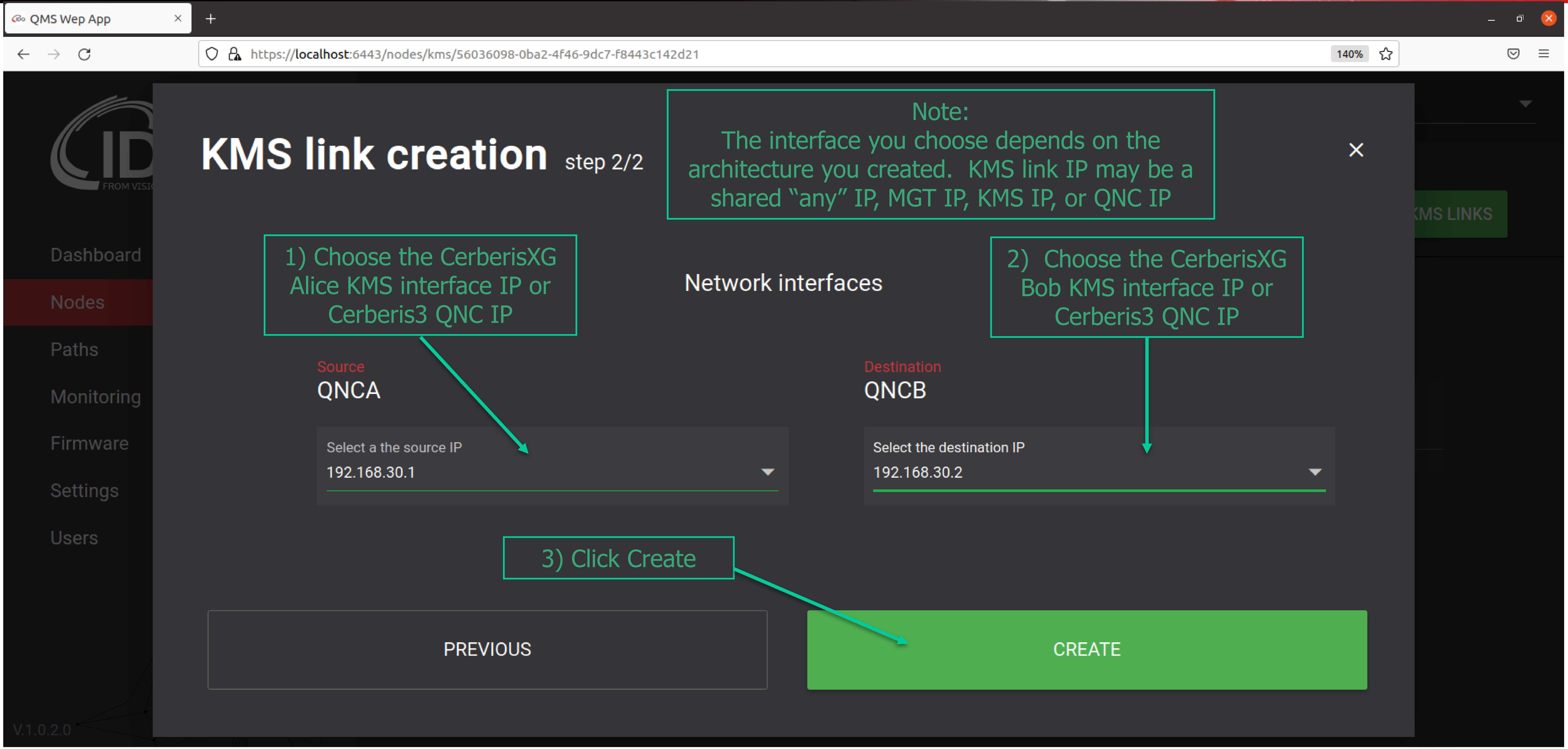
Create Node Links



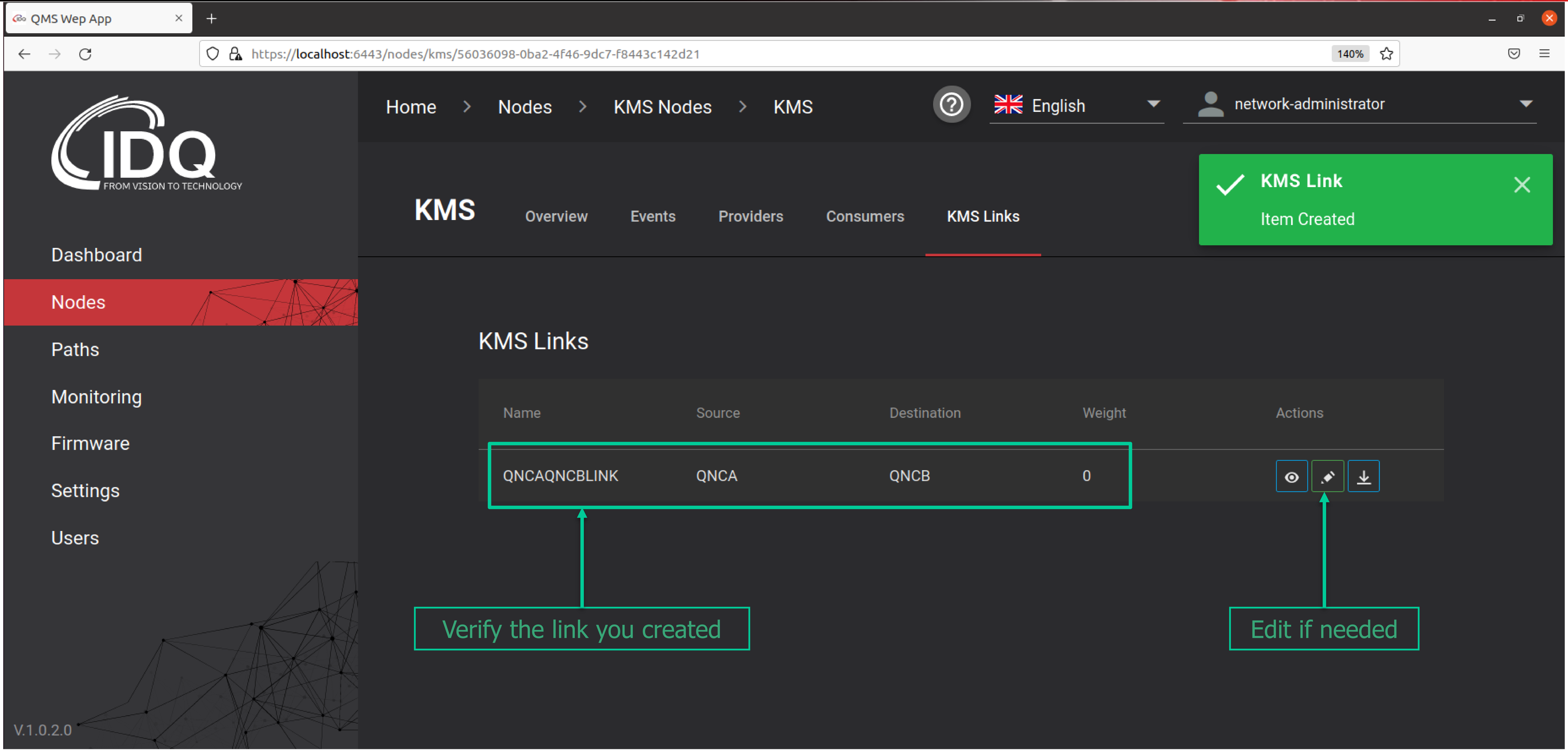
Create Node Links



Create Node Links



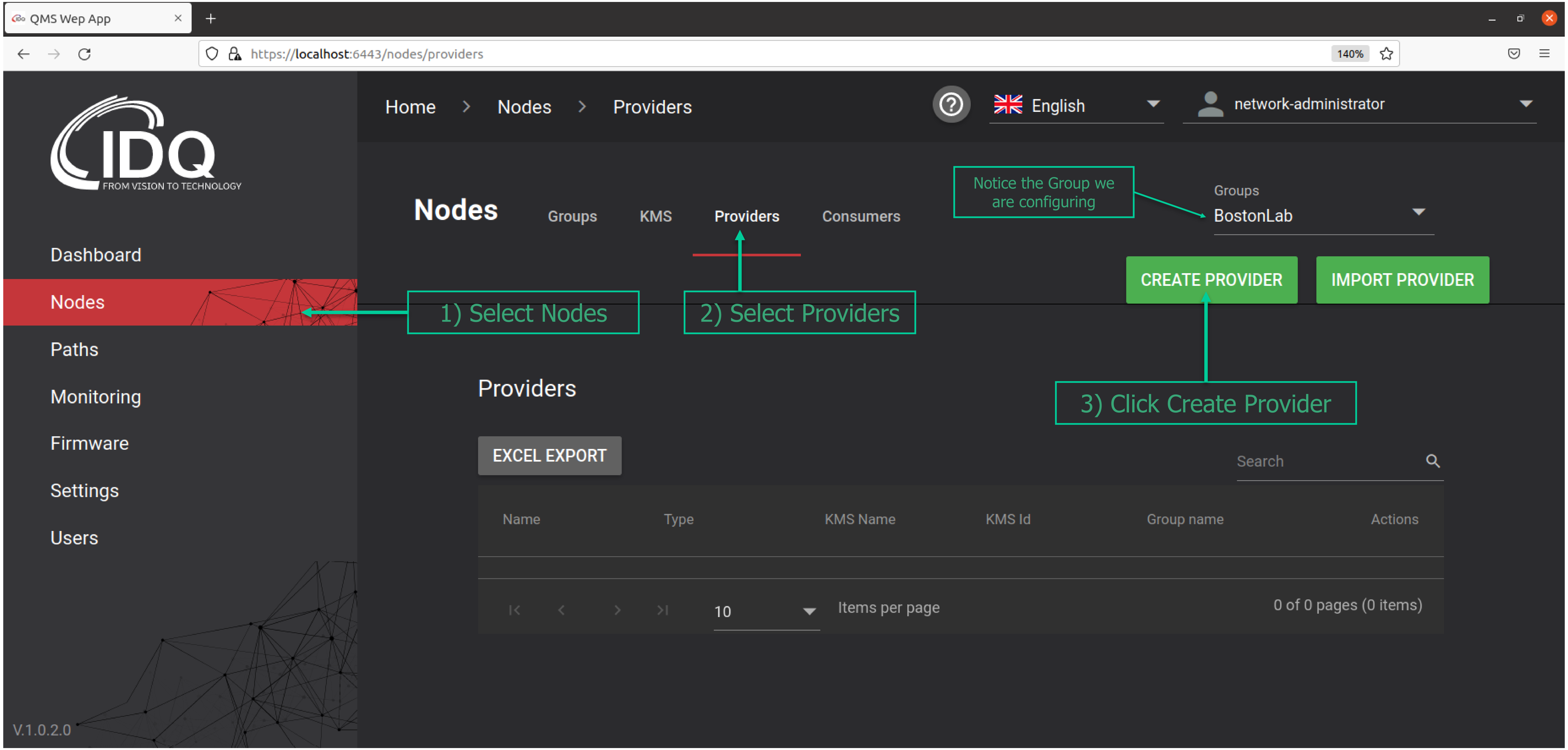
Create Node Links



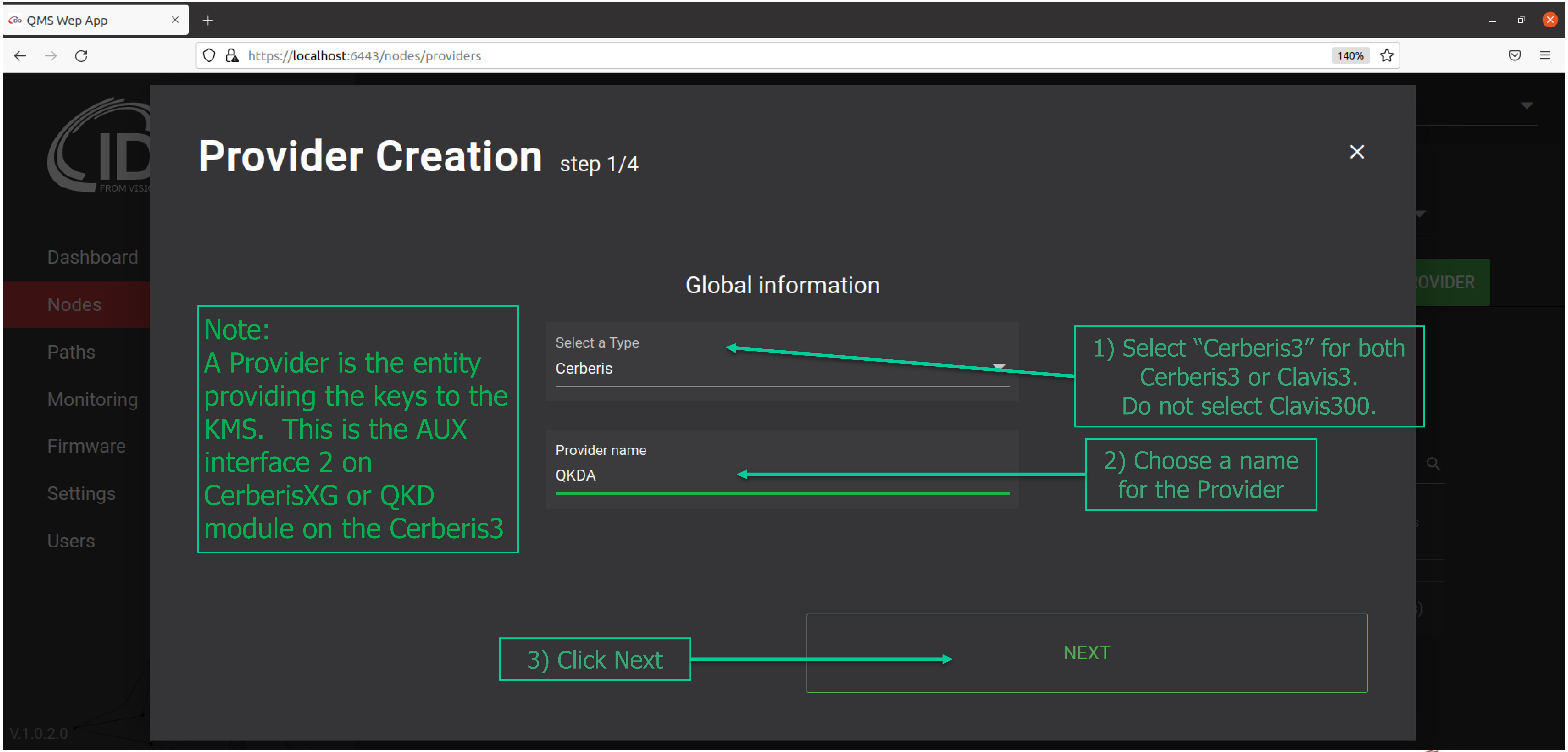
Create Path

Create Providers

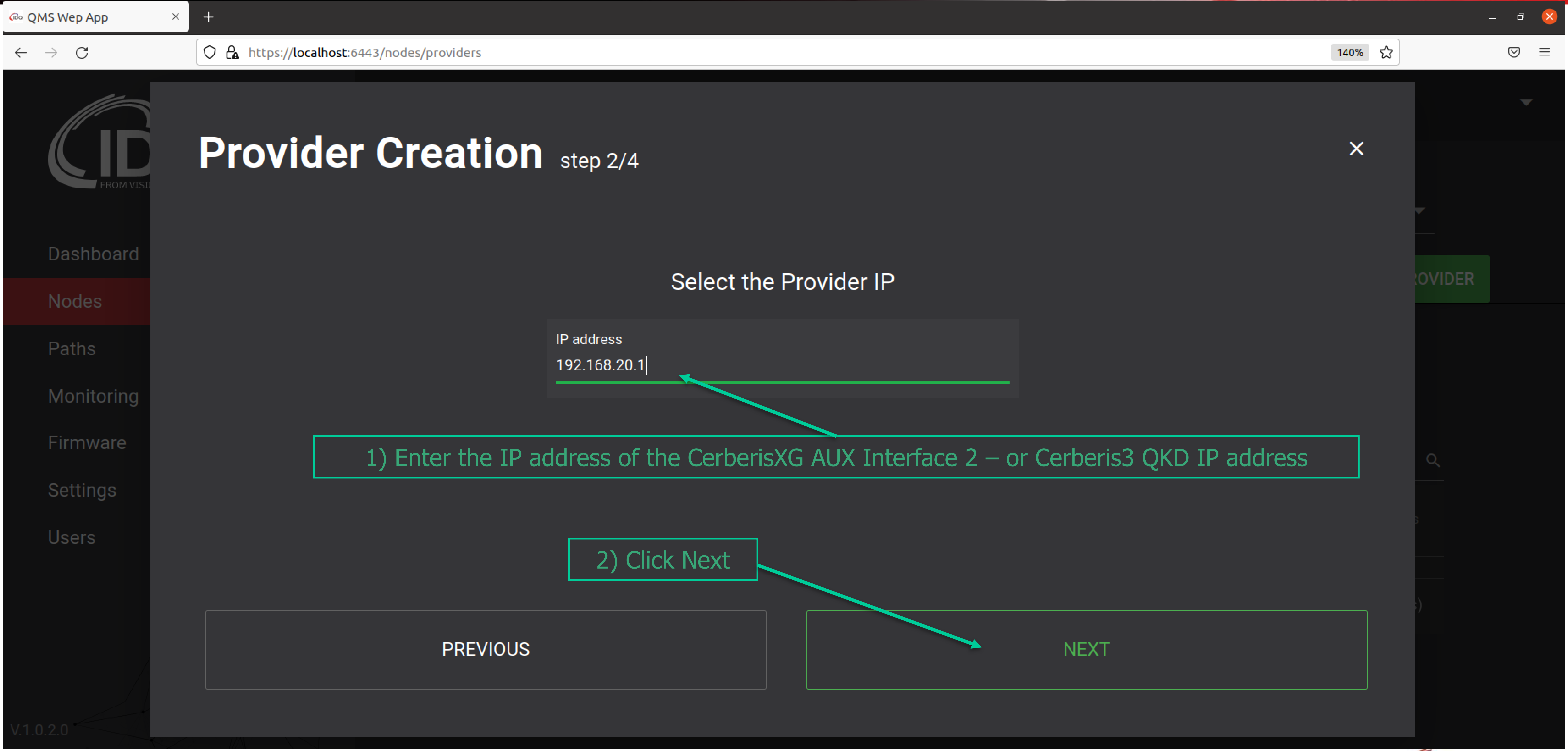
Create Providers



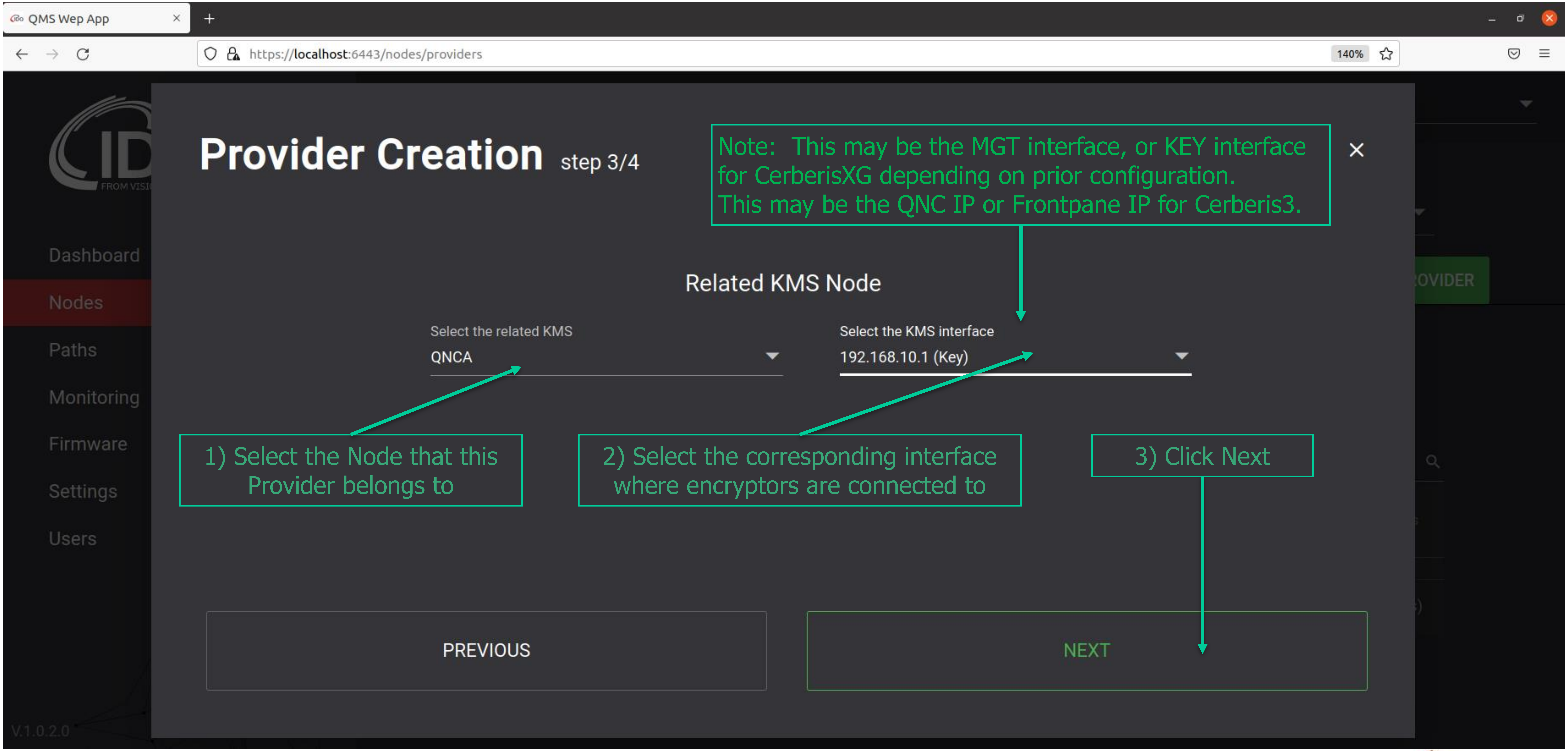
Create Providers



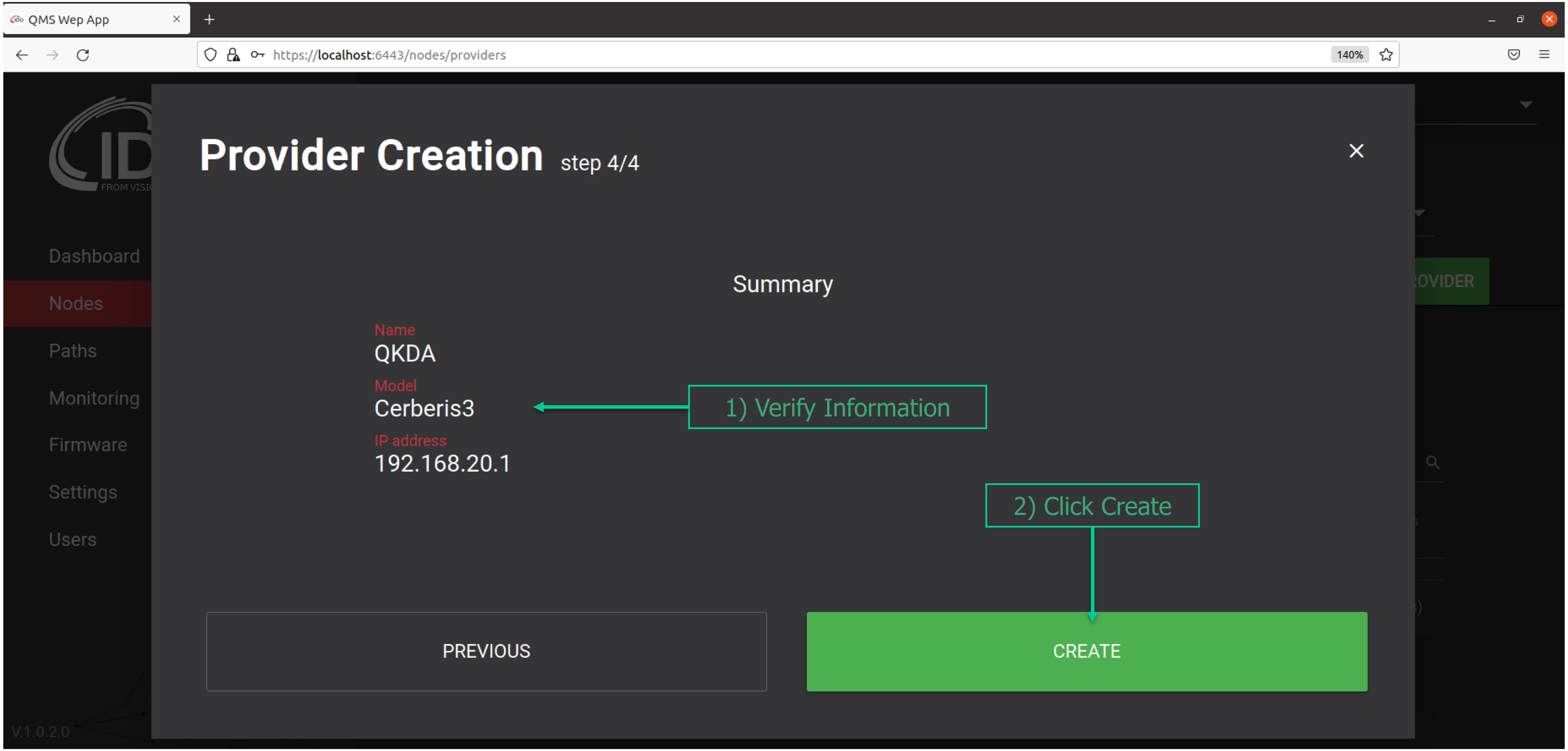
Create Providers



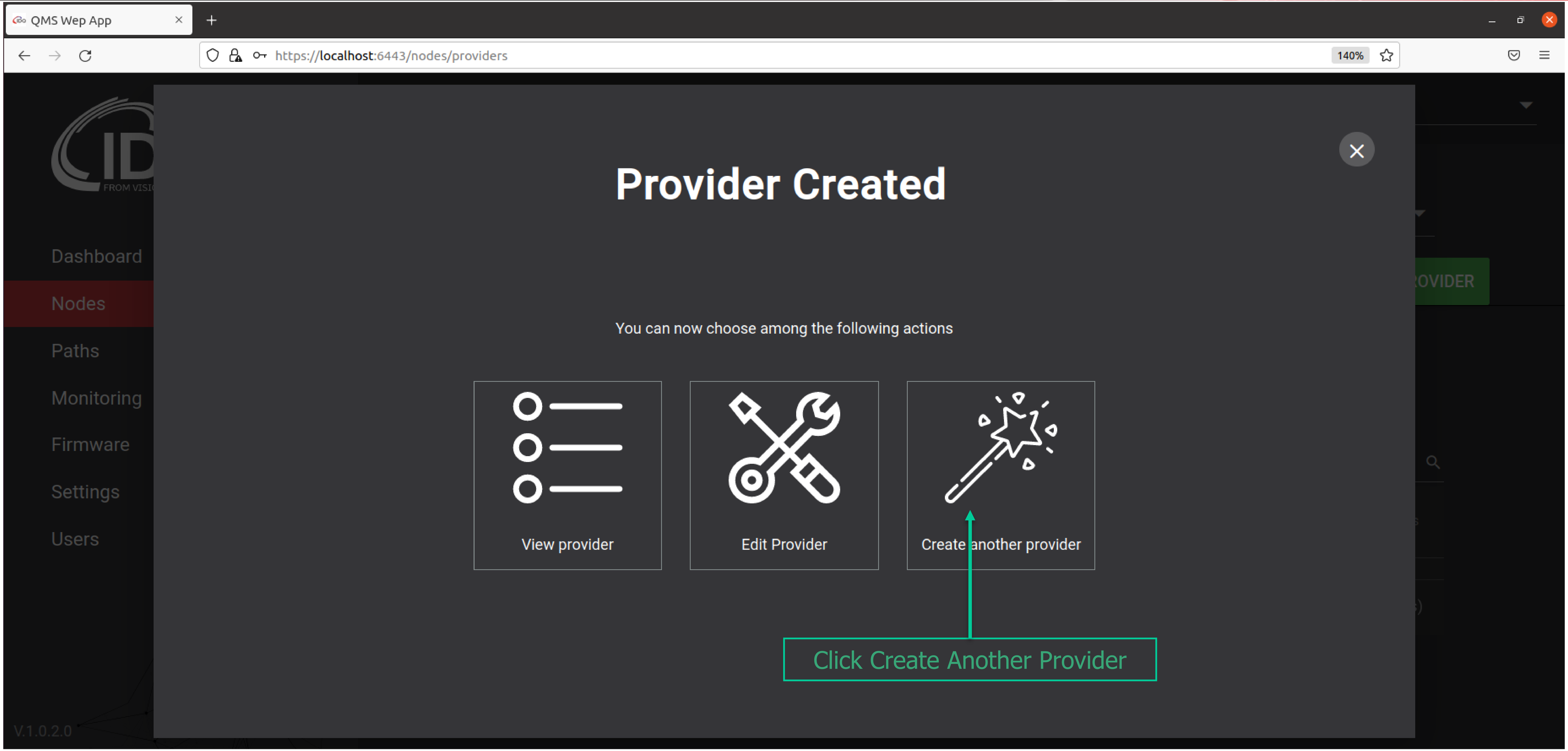
Create Providers



Create Providers

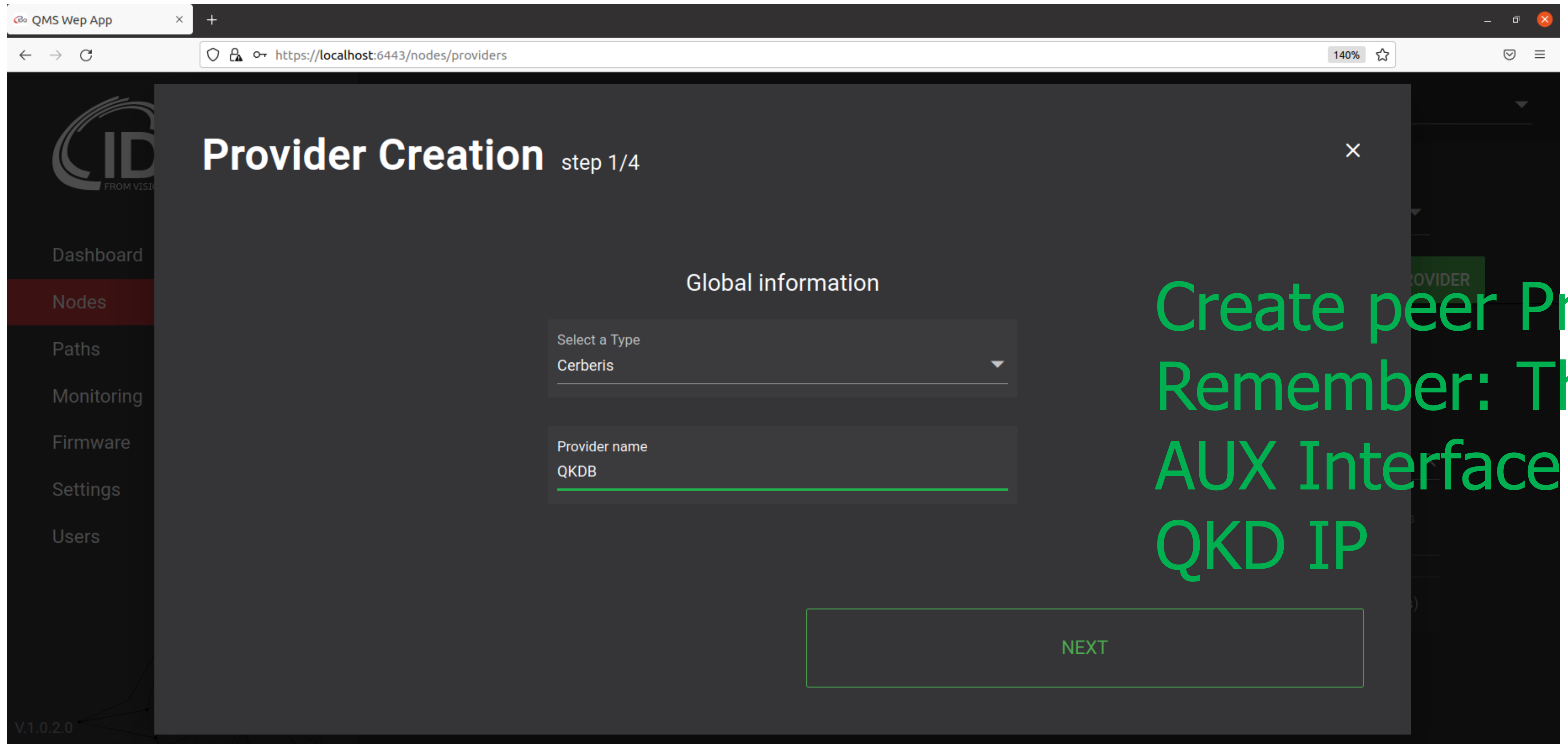


Create Providers

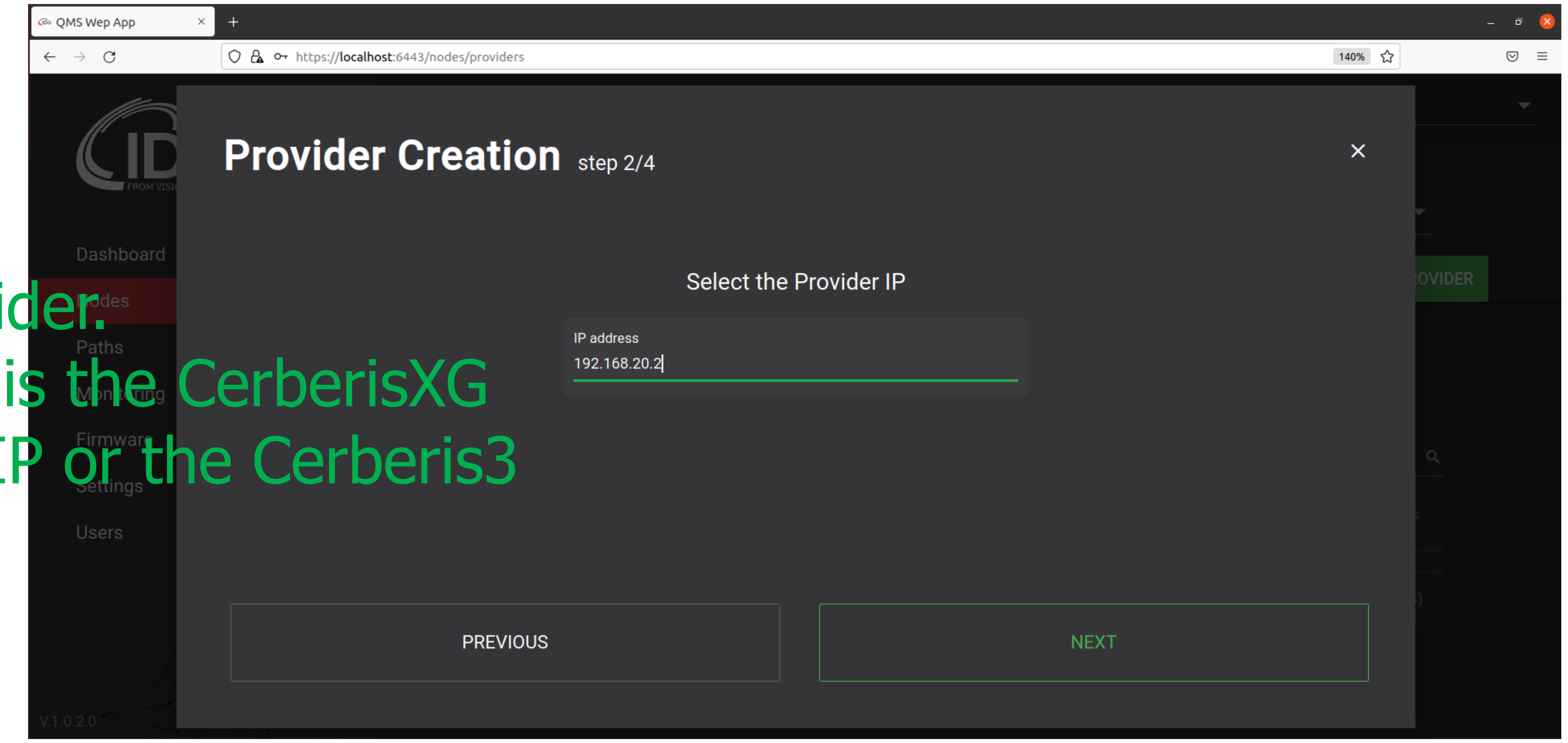


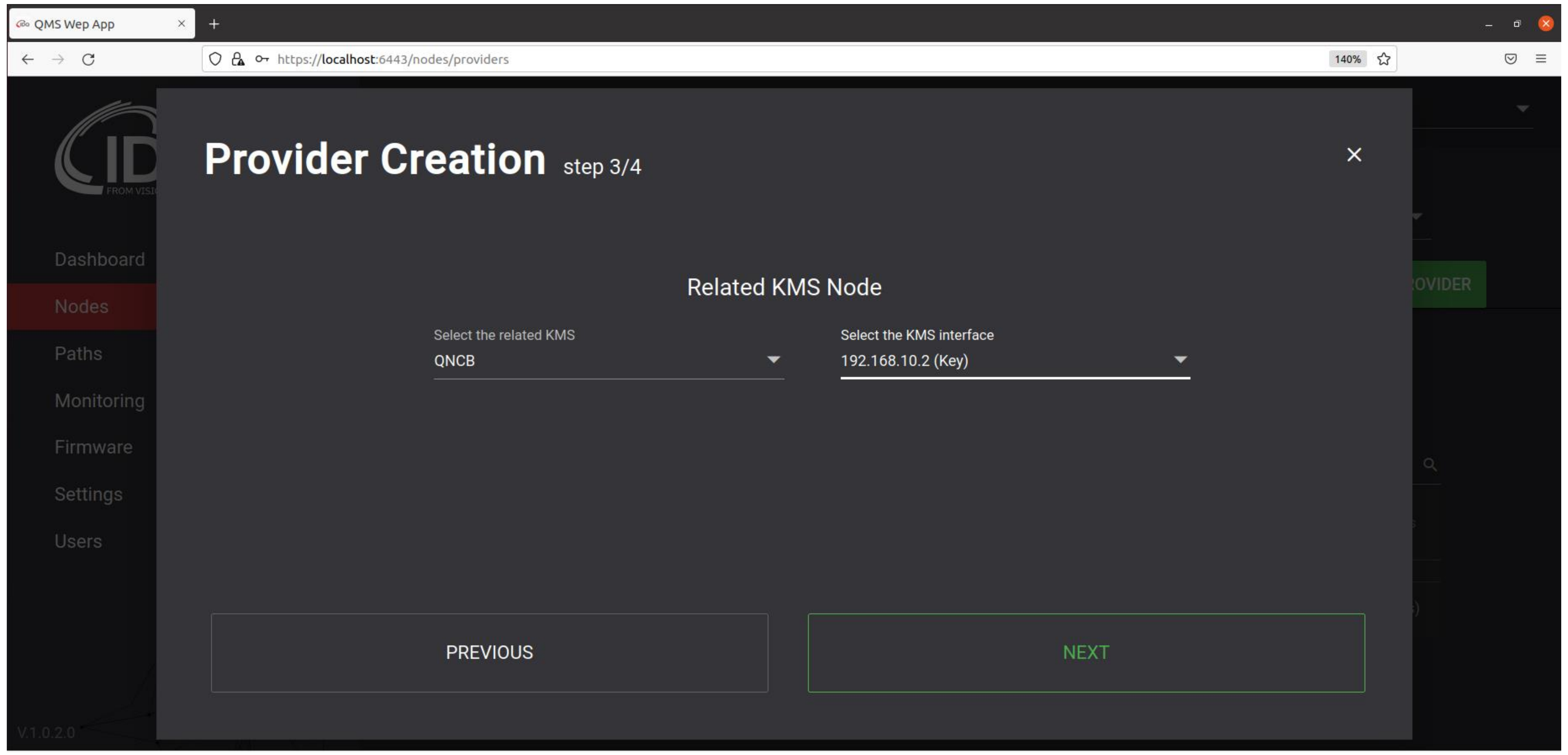
Repeat the same process to create the peer Provider

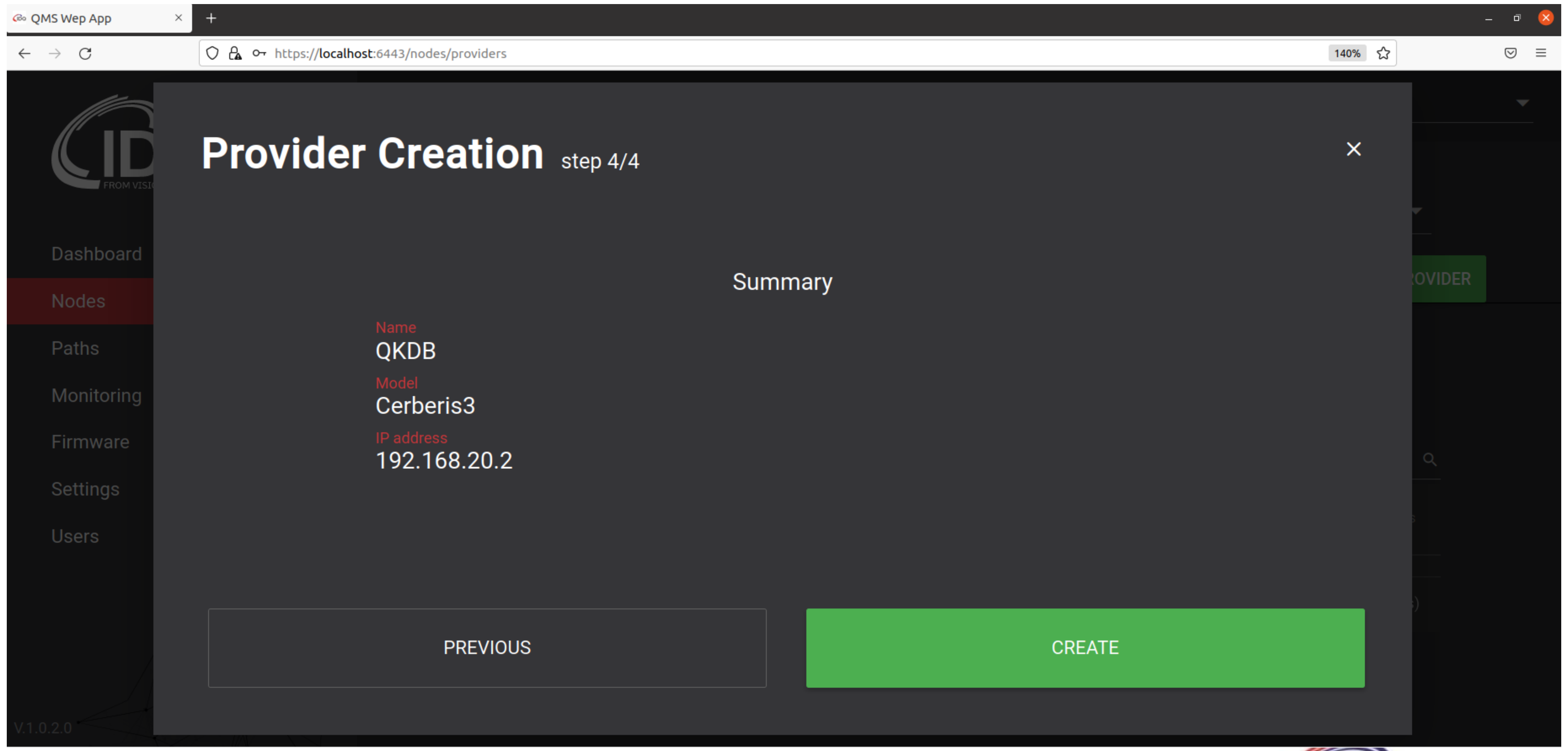
Create Providers



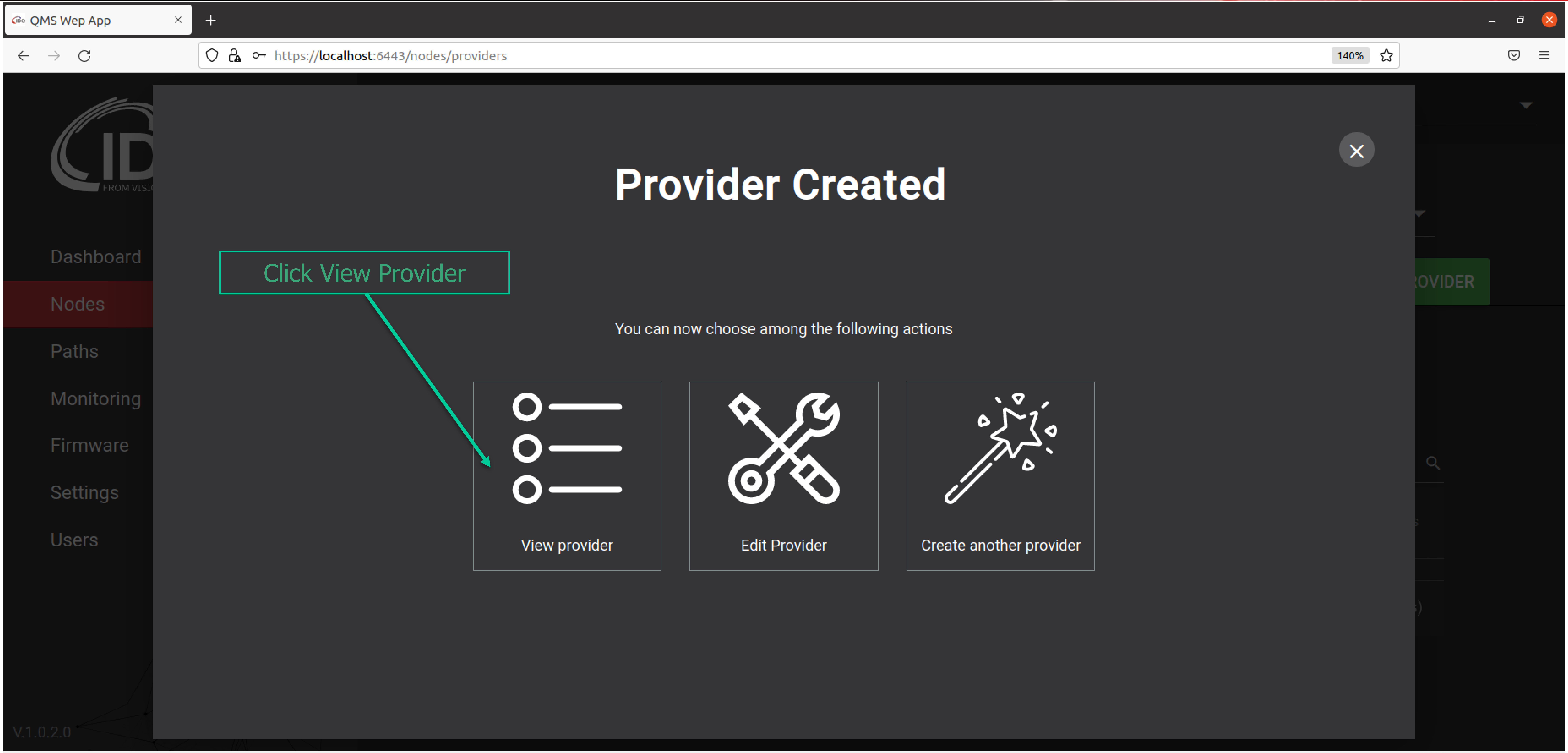
Create peer Provider.
Remember: This is the CerberisXG AUX Interface 2 IP or the Cerberis3 QKD IP



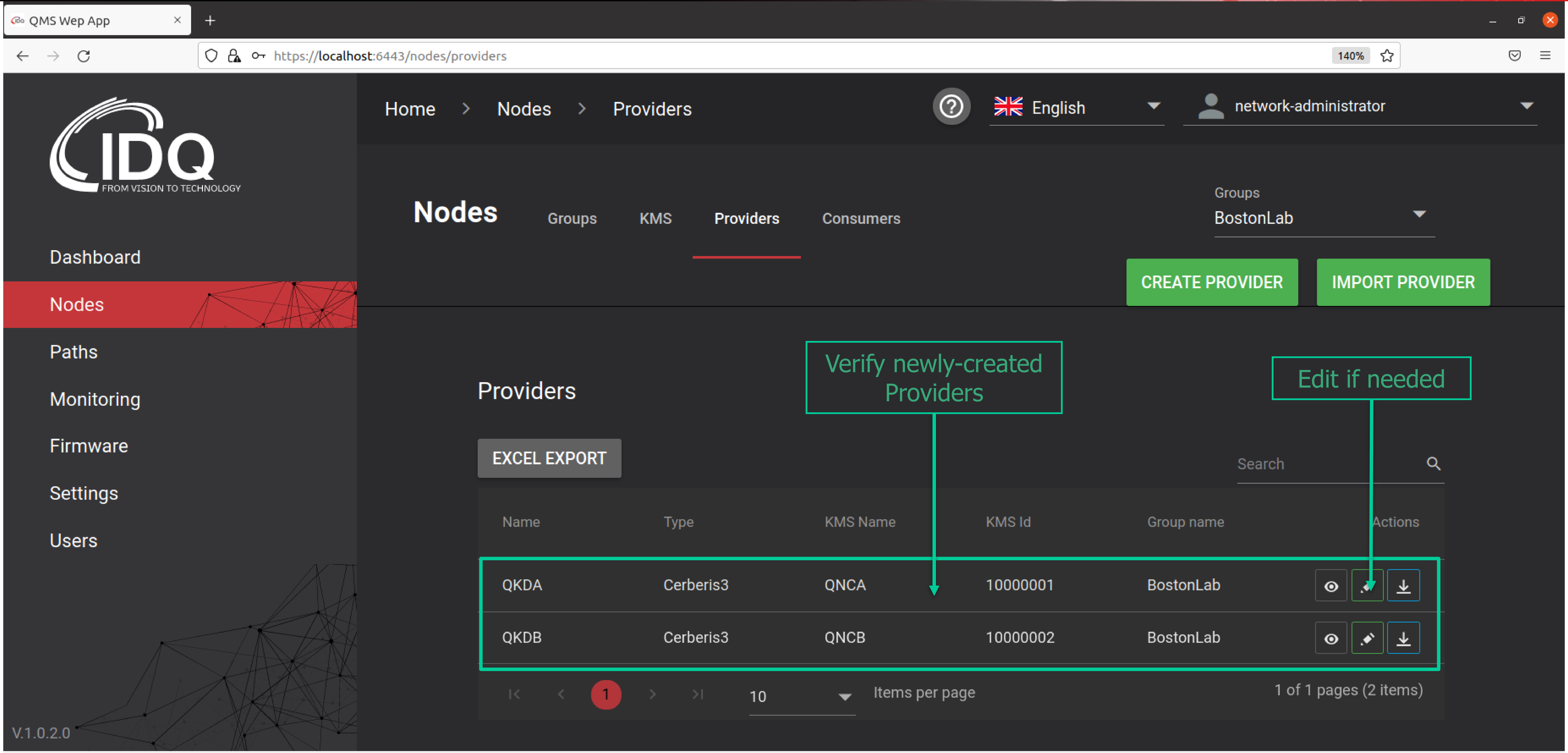




Create Providers

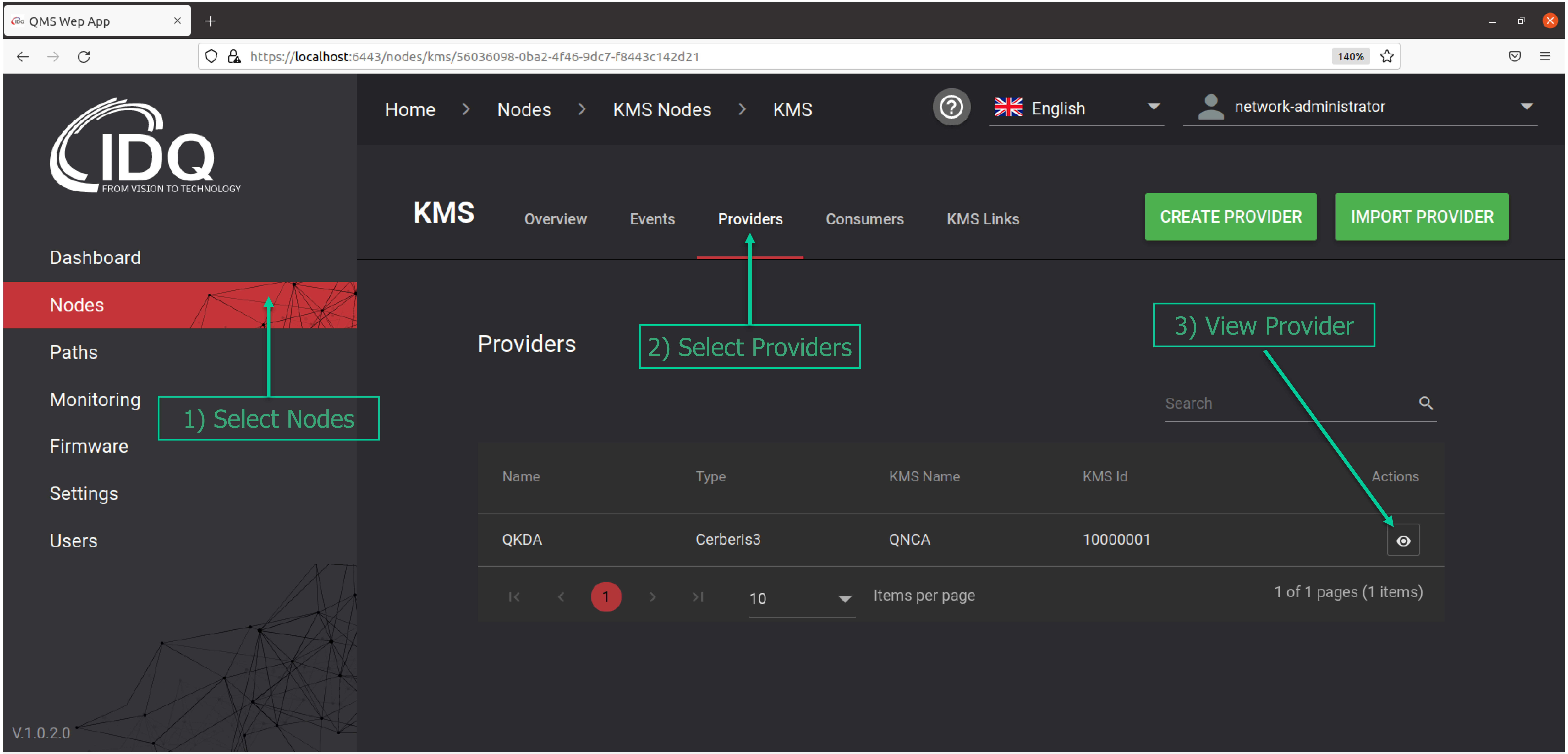


Create Providers

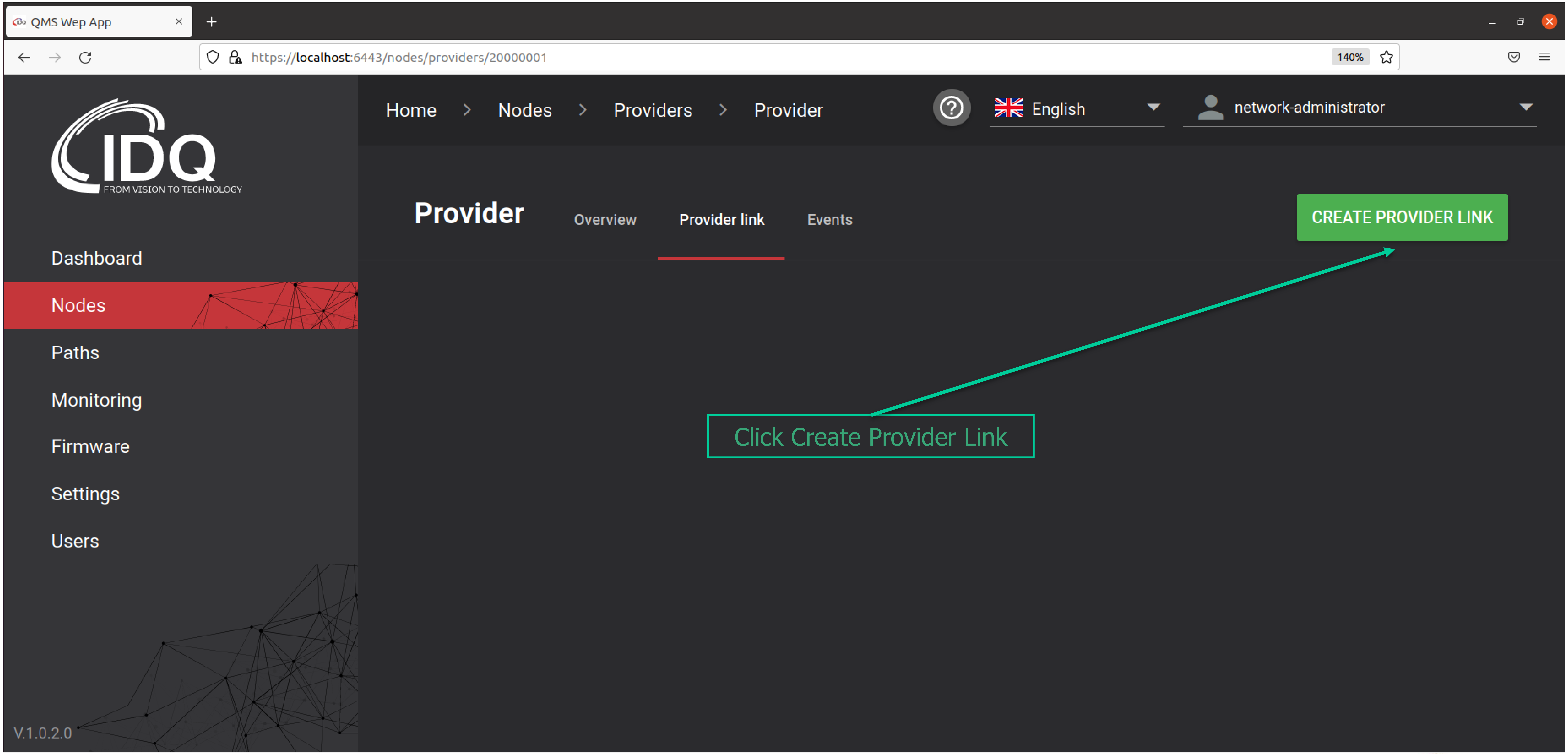


Create Provider Links

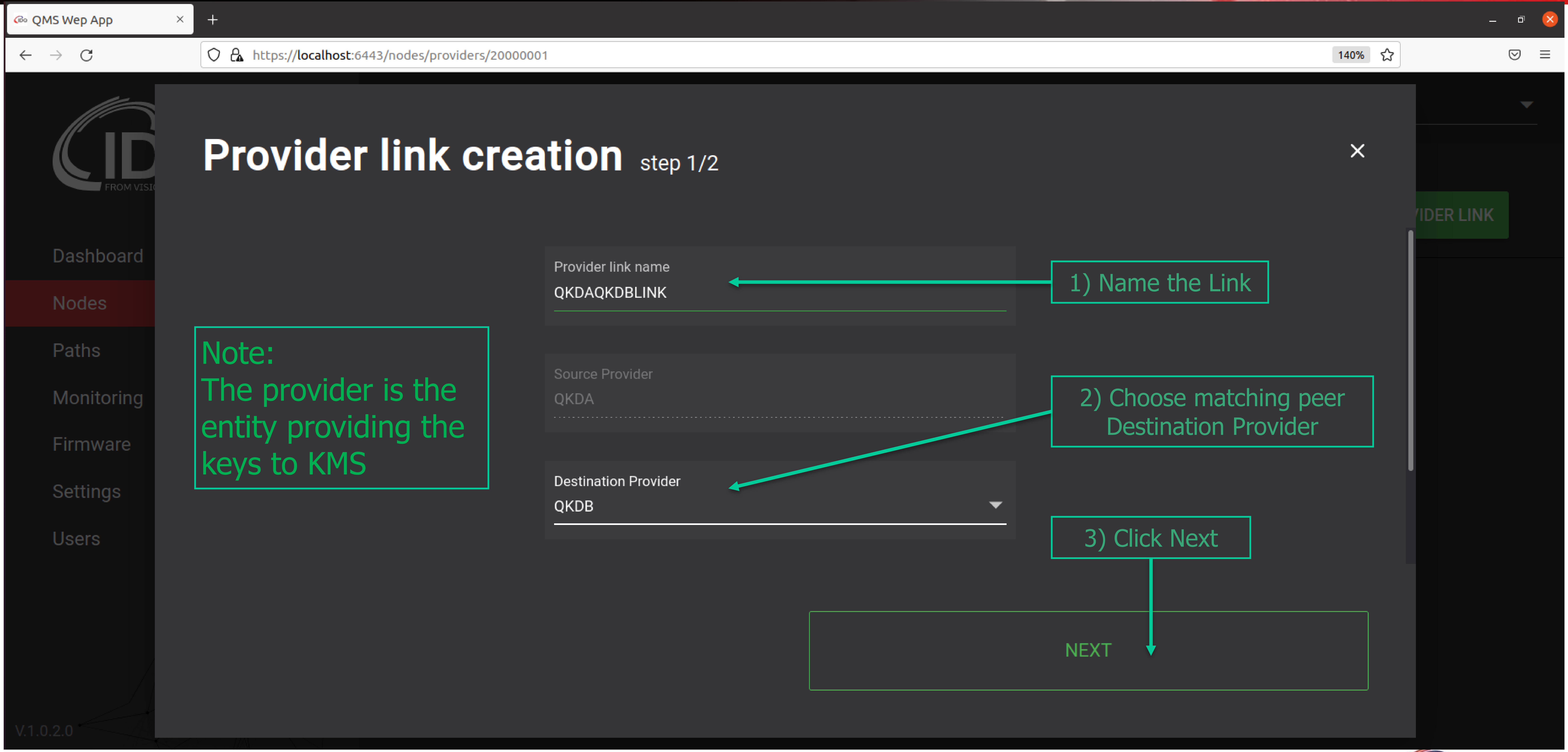
Create Provider Links



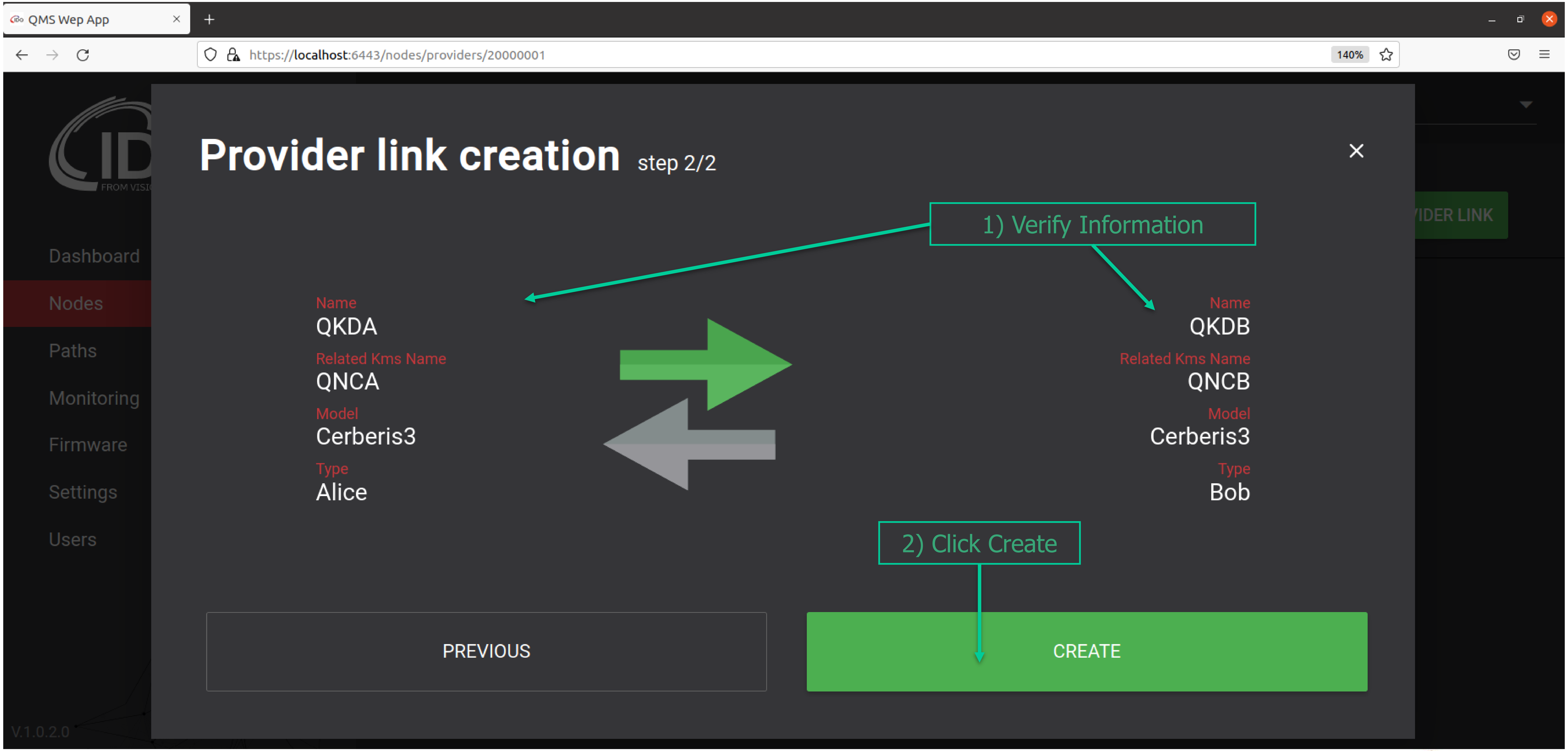
Create Provider Links



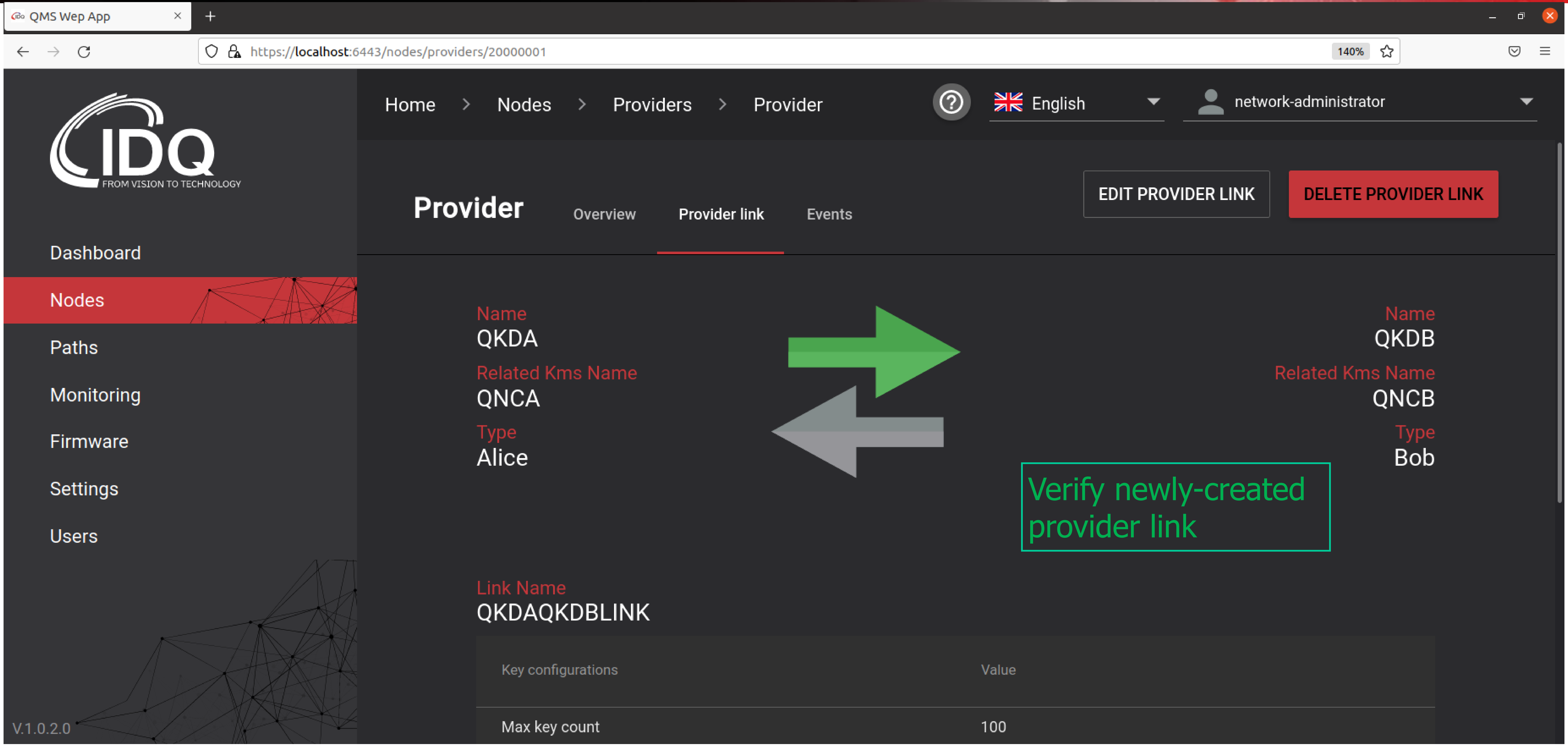
Create Provider Links



Create Provider Links

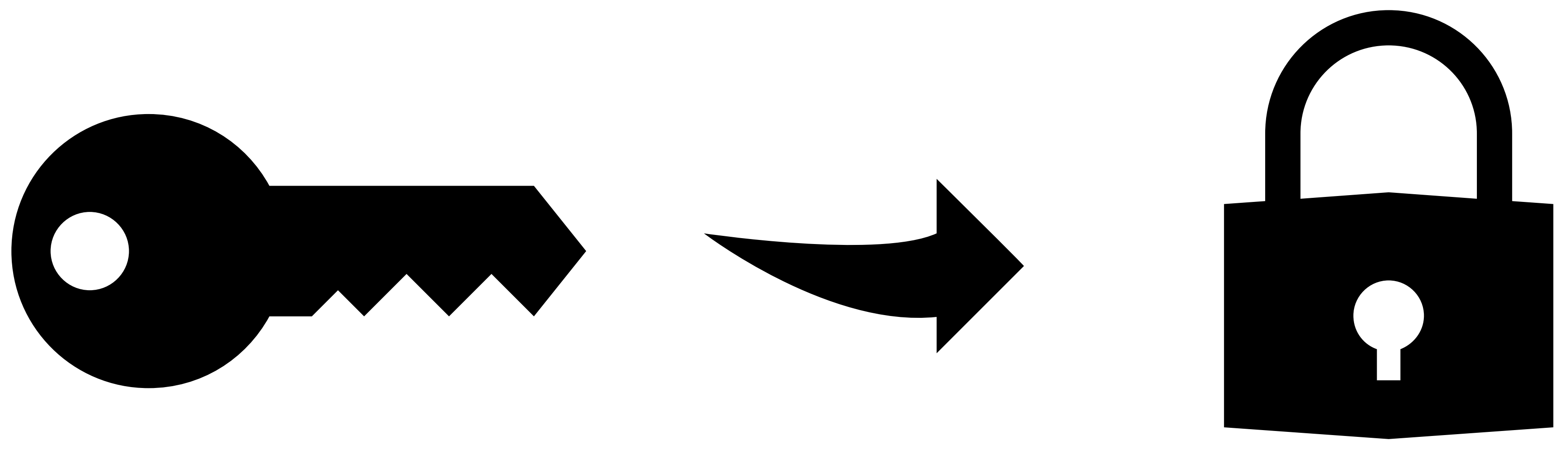


Create Provider Links

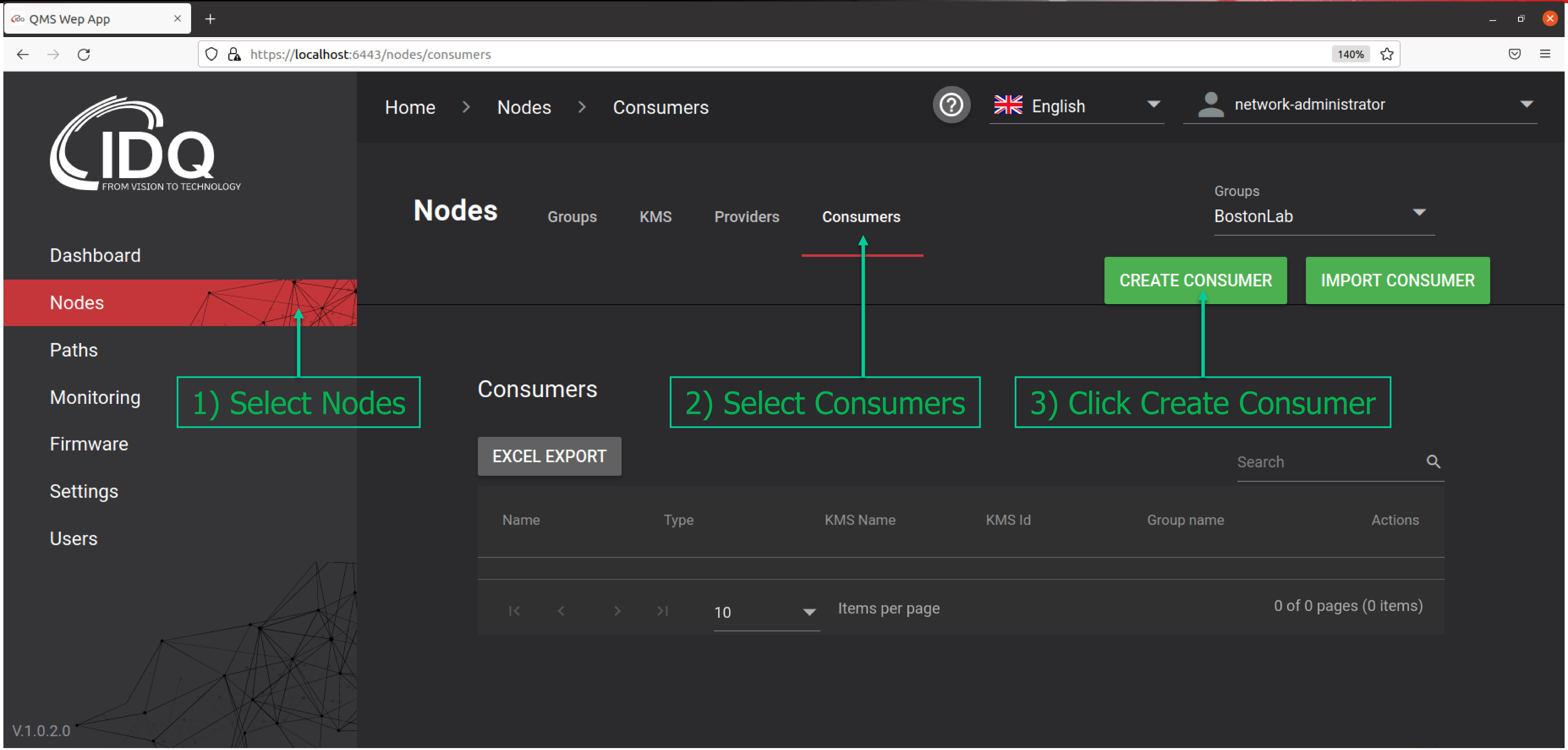


Create Path

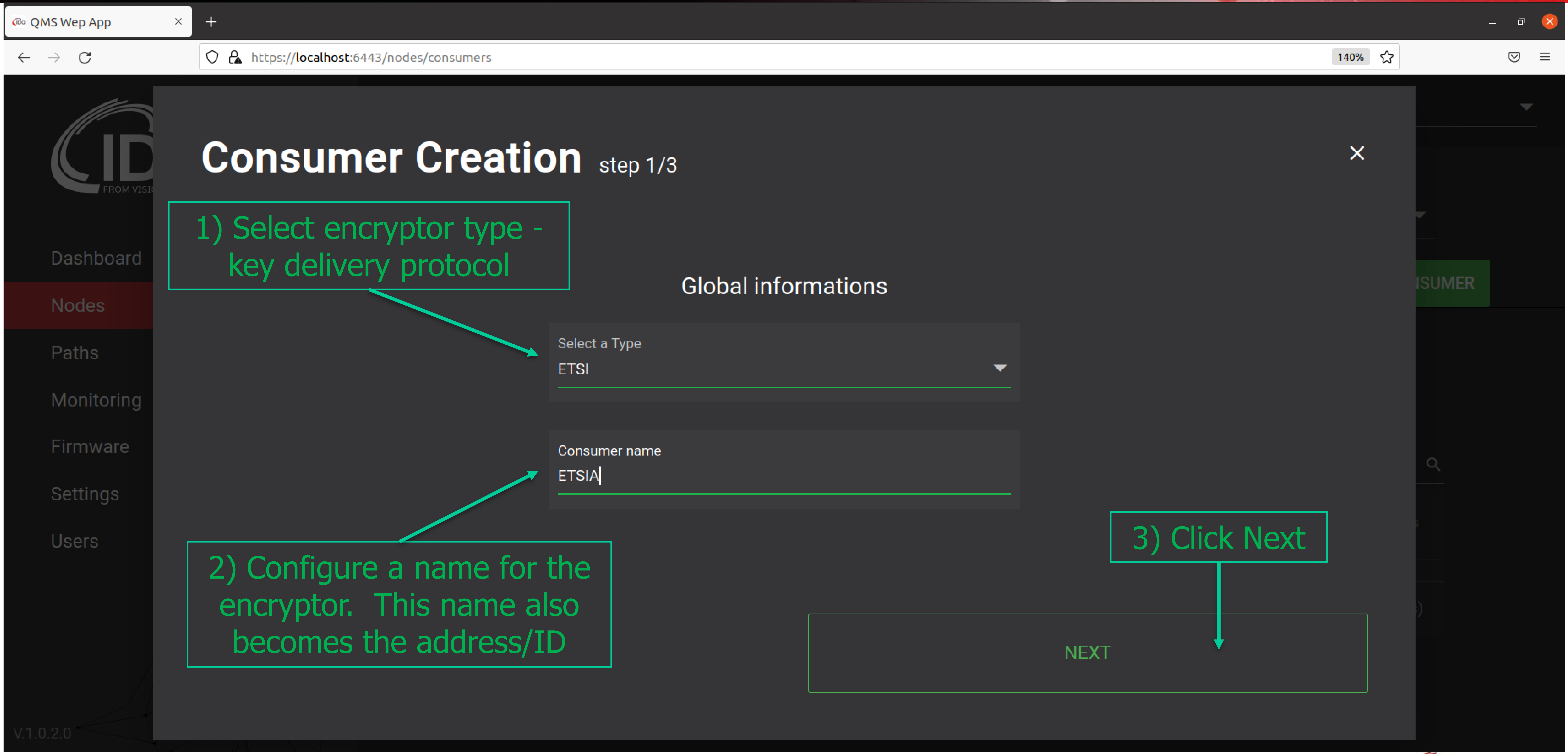
Create Consumer



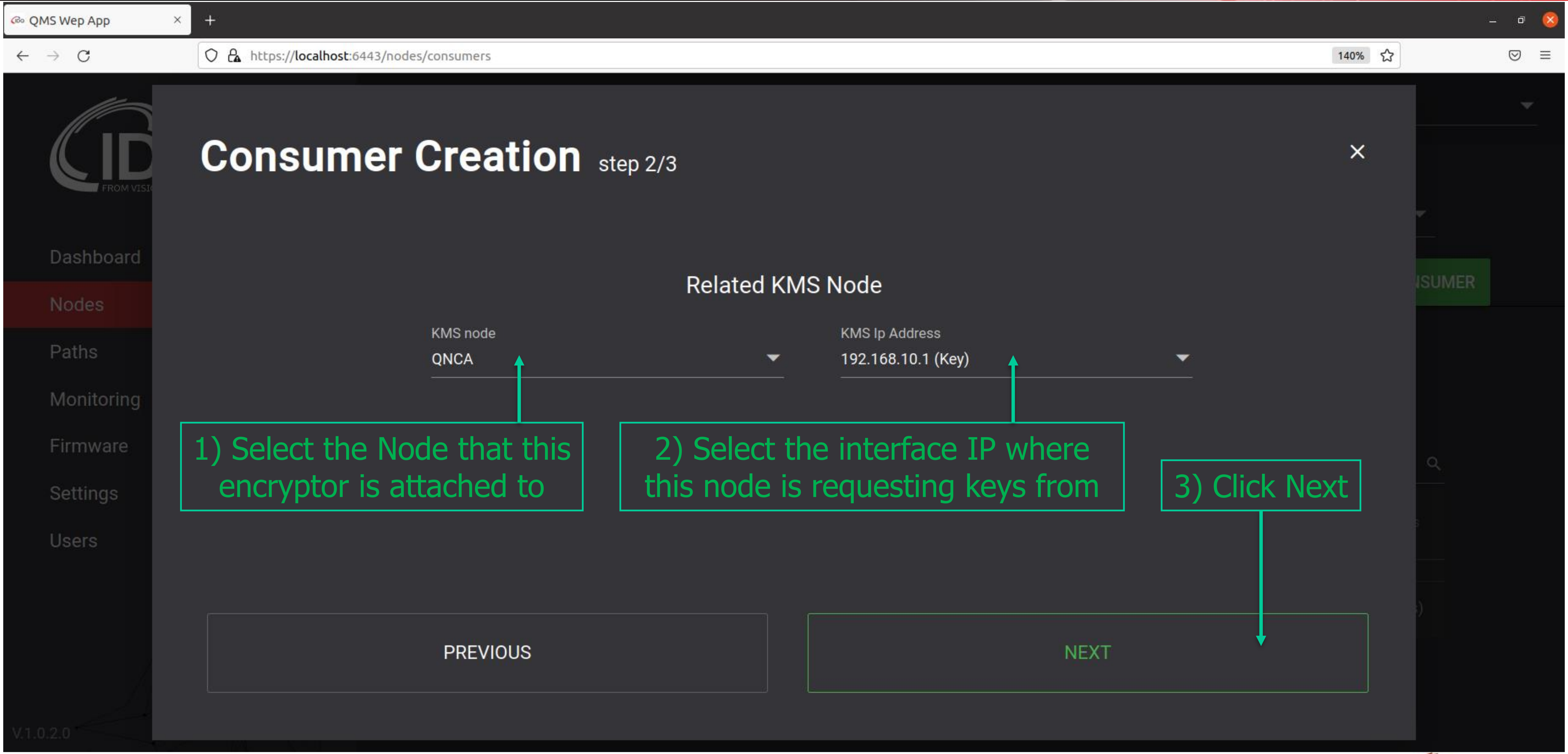
Create Consumer



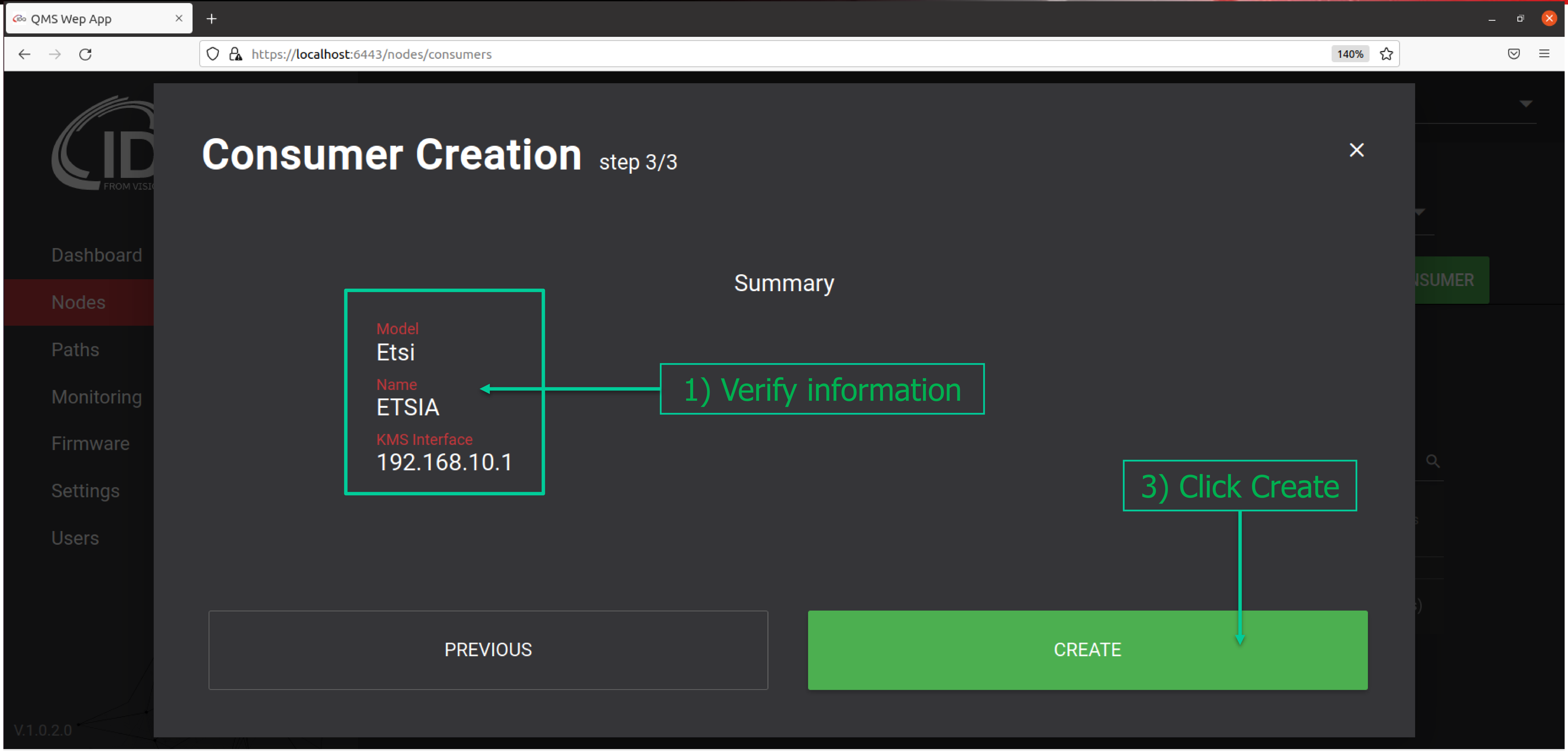
Create Consumer



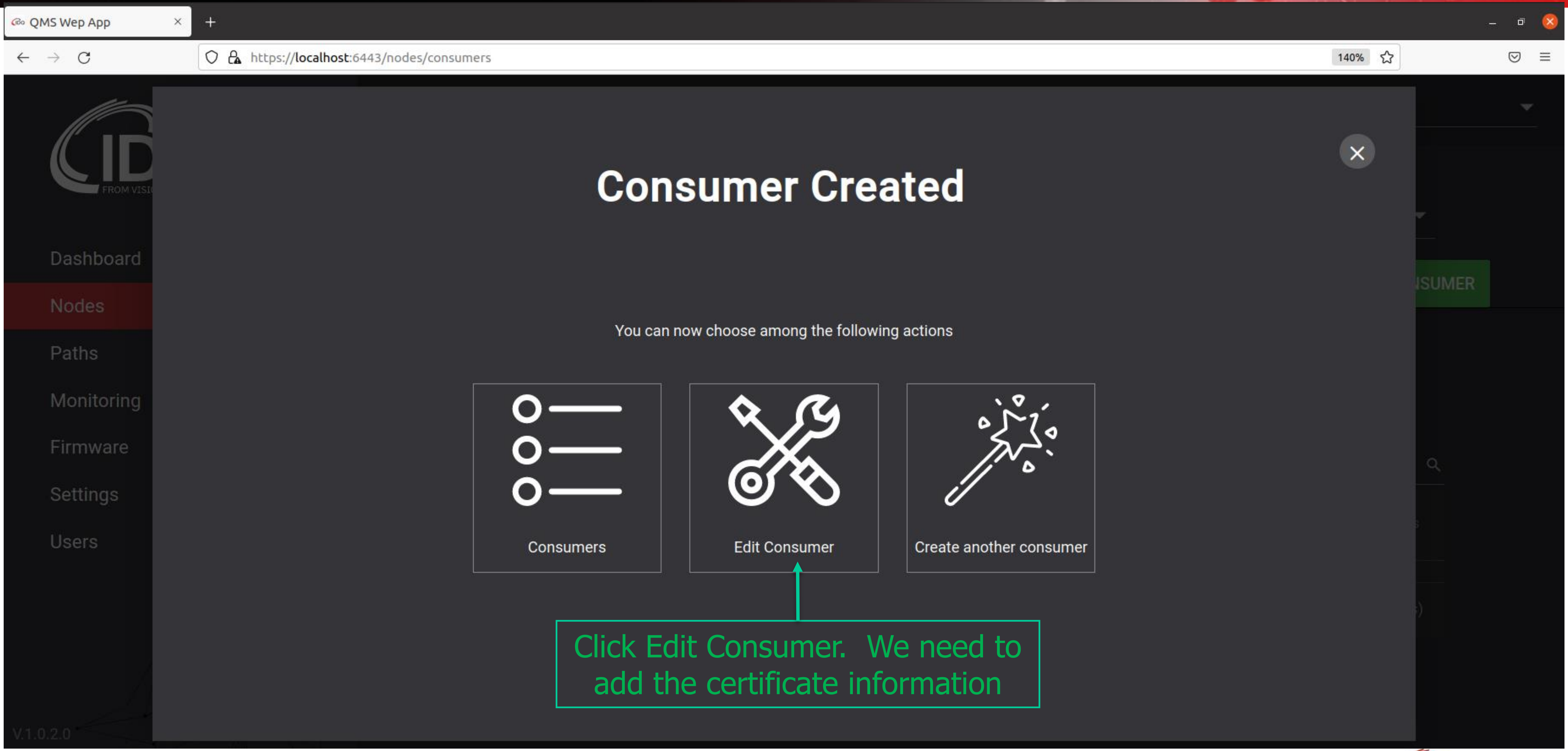
Create Consumer



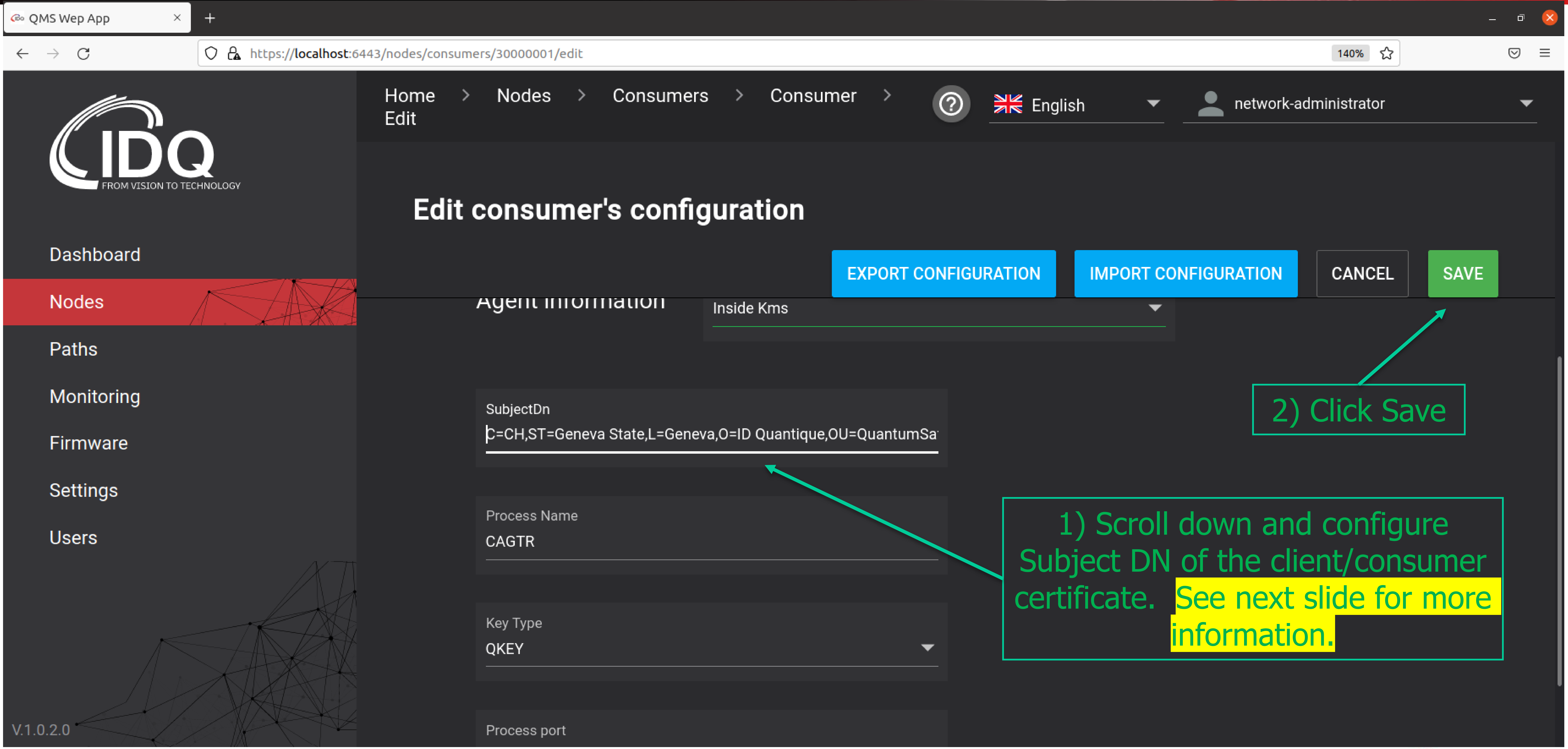
Create Consumer



Create Consumers



Create Consumers



Create Consumers

The SubjectDN is not needed for Cisco Encryptors.
To view the SubjectDN, open the certificate with openssl or other certificate management tool:

Example:
openssl x509 -in Client1.pem -text -noout

```
C:\Program Files\OpenSSL-Win64\bin>openssl.exe x509 -in "C:\Users\Public\Documents\Cert\Files\enc-bob.pem" -noout -text
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            11:5f:fa:6a:2d:be:c8:6f:c0:e7:9d:ea:69:17:2e:cc:3d:5e:9d:03
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C = CH, ST = Geneva State, L = Geneva, O = ID Quantique, OU = Security
        Validity
            Not Before: Mar  3 14:14:00 2020 GMT
            Not After : Mar  3 14:14:00 2023 GMT
        Subject: C = CH, ST = Geneva State, L = Geneva, O = ID Quantique, OU = QuantumSafe, CN = enc-bob
        Subject Public Key Info:
            Public Key Algorithm: id-ecPublicKey
                Public-Key: (256 bit)
                pub:
                    04:0b:f4:79:8e:a7:fd:c3:32:82:4a:97:96:71:11:
                    cf:35:13:f9:ff:37:d5:12:43:cd:dc:e8:bf:5c:d2:
                    5e:4f:39:84:95:5c:67:da:26:3b:f1:dc:97:89:36:
                    80:6d:d9:de:43:41:69:f6:67:3f:f8:fd:ee:b0:cf:
                    f6:1d:53:d3:b1
                ASN1 OID: prime256v1
                NIST CURVE: P-256
        X509v3 extensions:
```

Please see next slide for additional important information

Create Consumers

SubjectDn

- The SubjectDN field is not needed for Cisco consumer configuration.

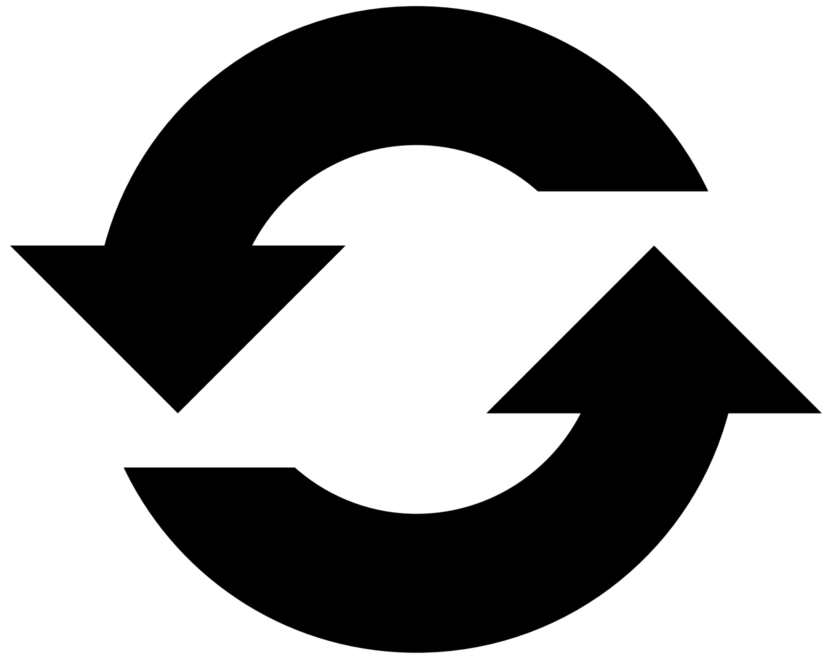
- Copy the Subject from the encryptor's client certificate. Make sure you remove any spaces between delimiters. The delimiter must be changed to a comma "," if another delimiter is displayed.
- A space inside one of the attributes is valid. Not valid between delimiters.

Example: openSSL ouput:

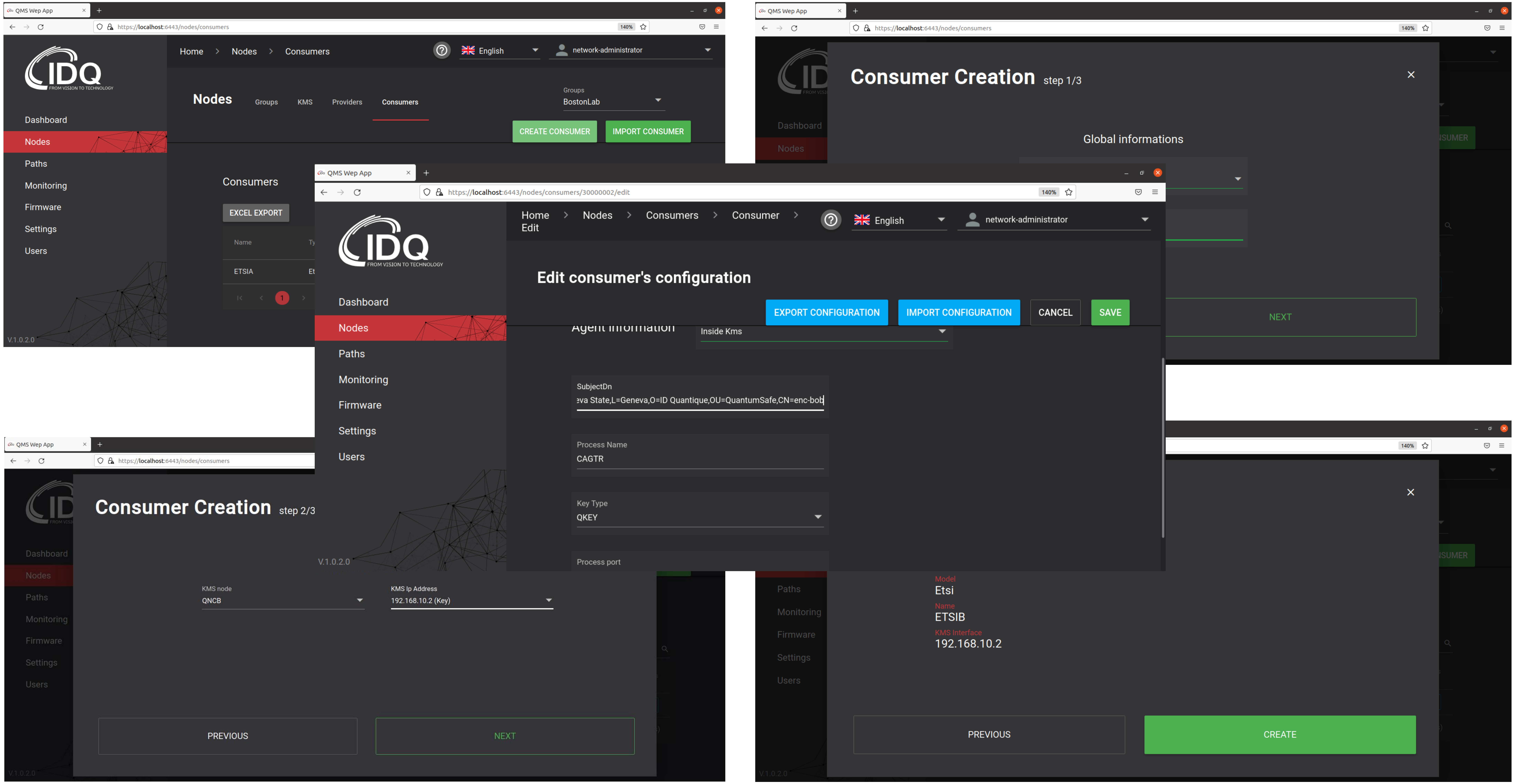
C = CH, ST = Geneva State, L = Geneva, O = ID Quantique, OU = QuantumSafe, CN = enc-bob
This is **NOT CORRECT** (we have **spaces**), see yellow portions
Space on the O attribute (ID QUANTIQUE) may have a space

You need to import **the following Subject DN into QMS**
C=CH,ST=Geneva State,L=Geneva,O=ID Quantique,OU=QuantumSafe,CN=enc-bob

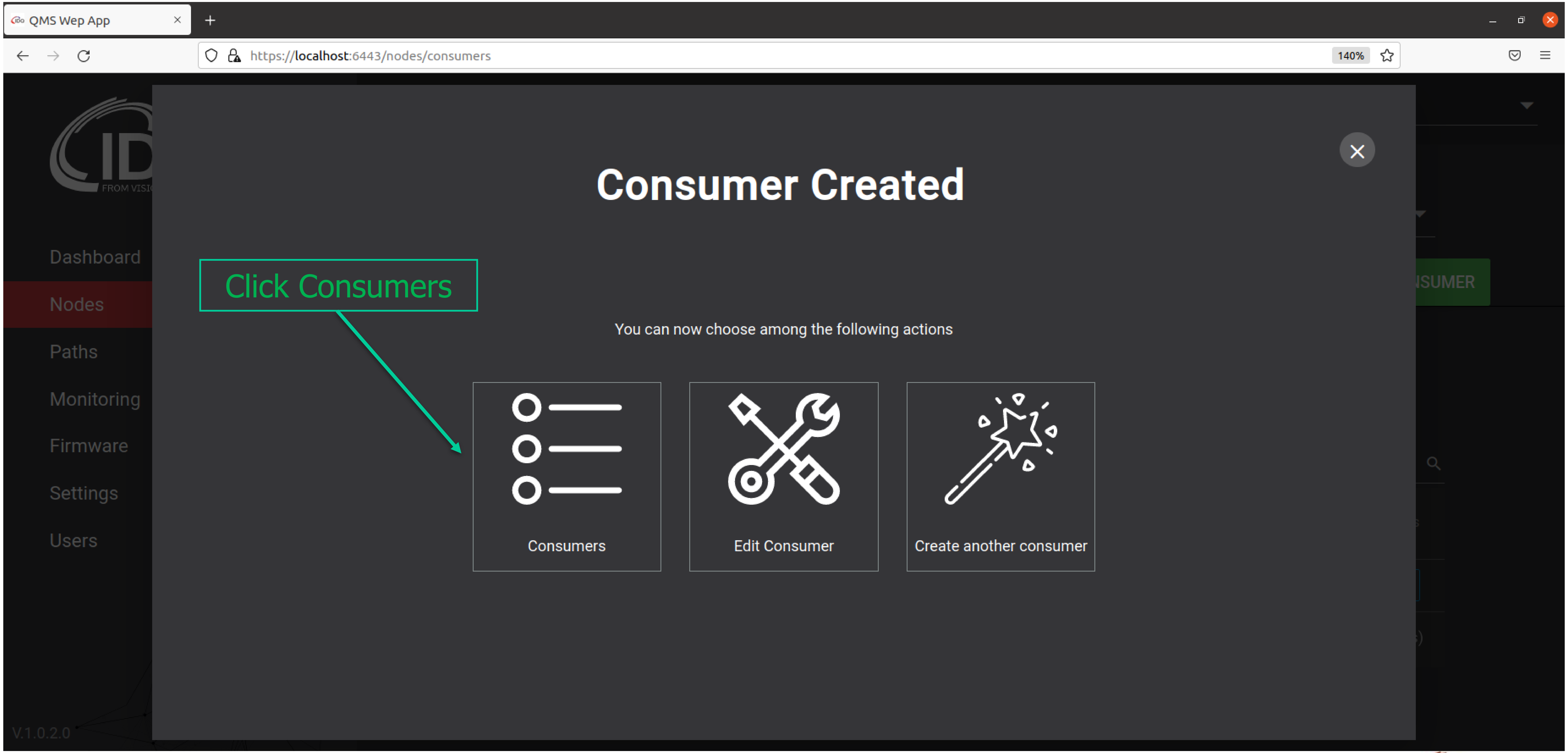
Repeat the same process to create the peer Consumer



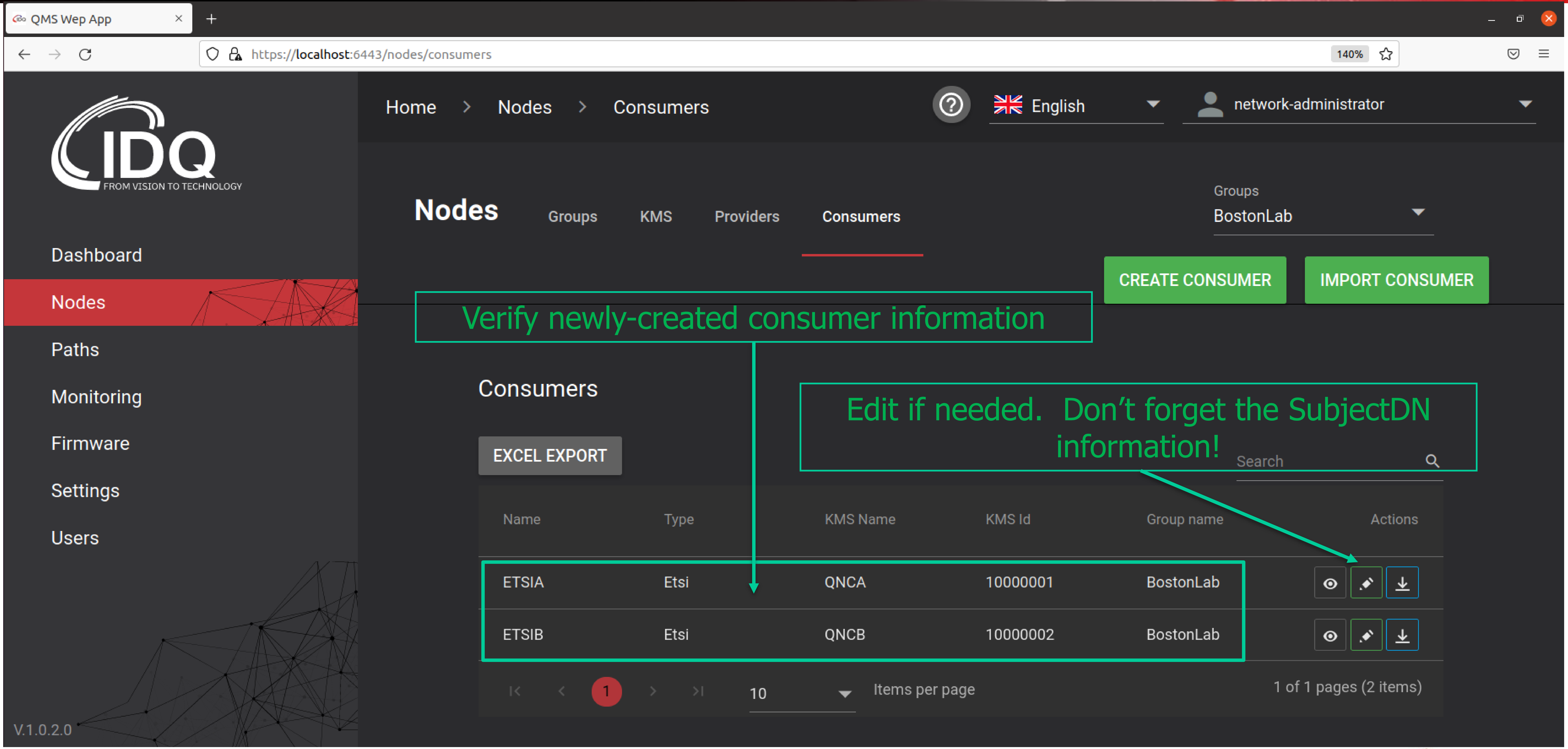
Create Consumers



Create Consumers

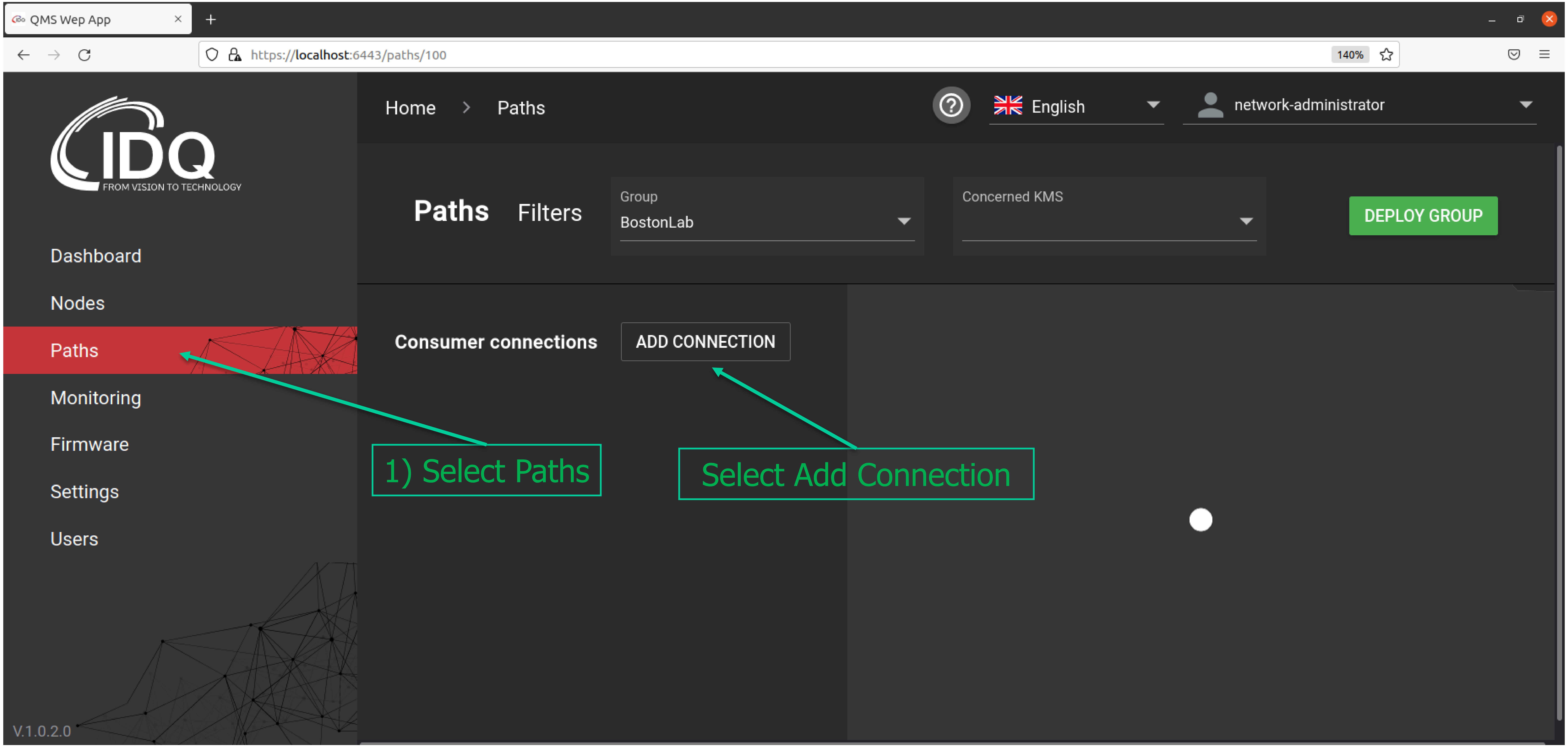


Create Consumers

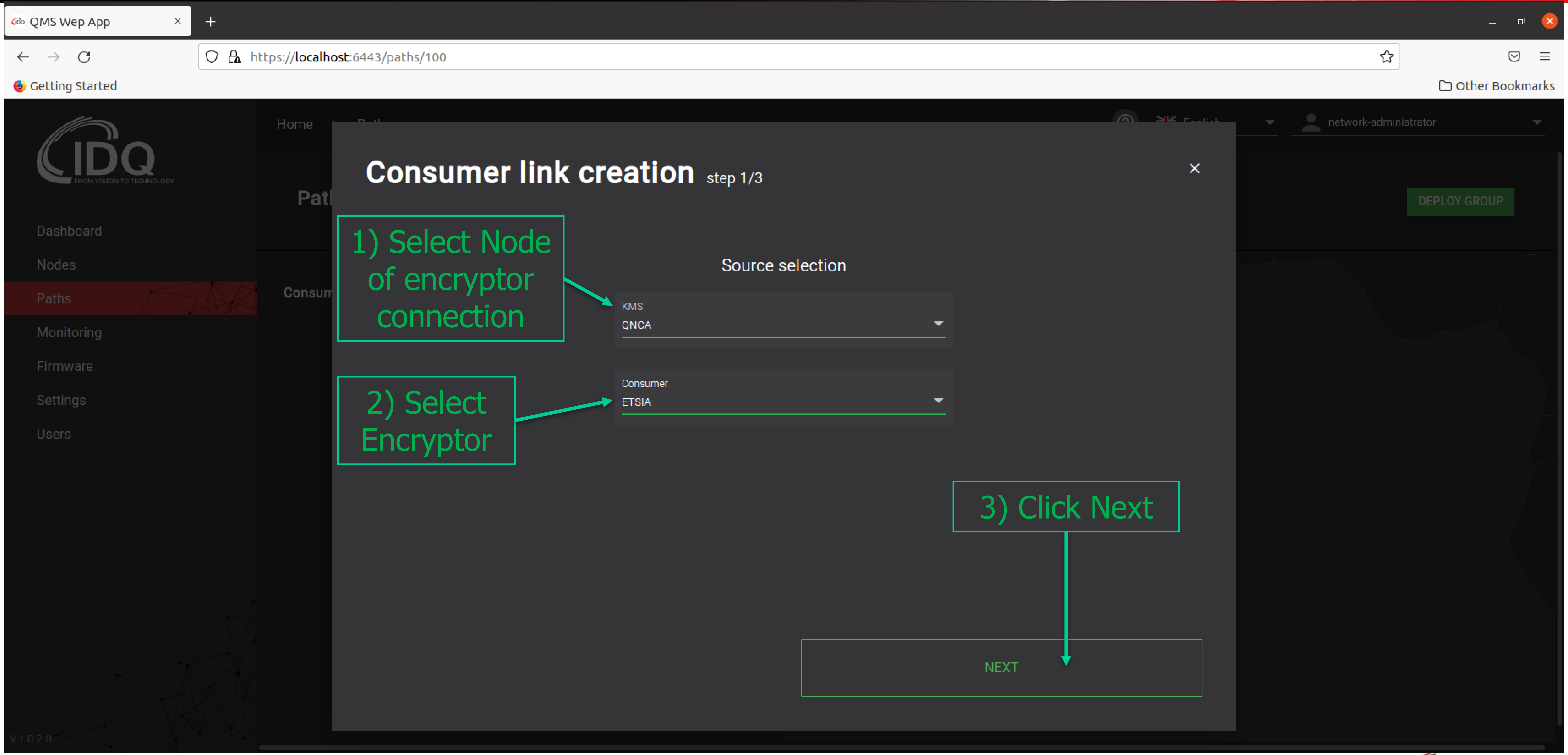


Create Path

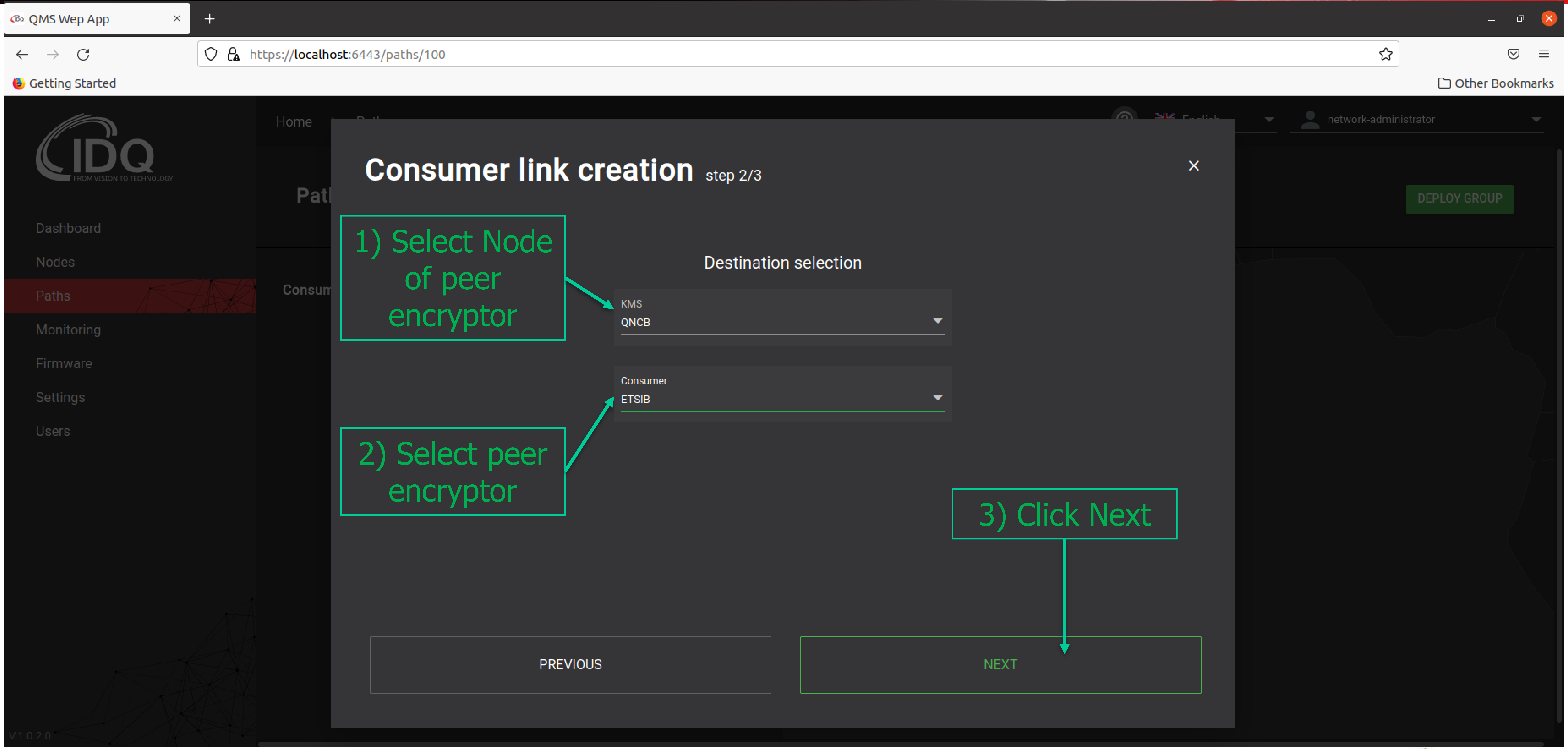
Create Consumer Links and Key Routing Path



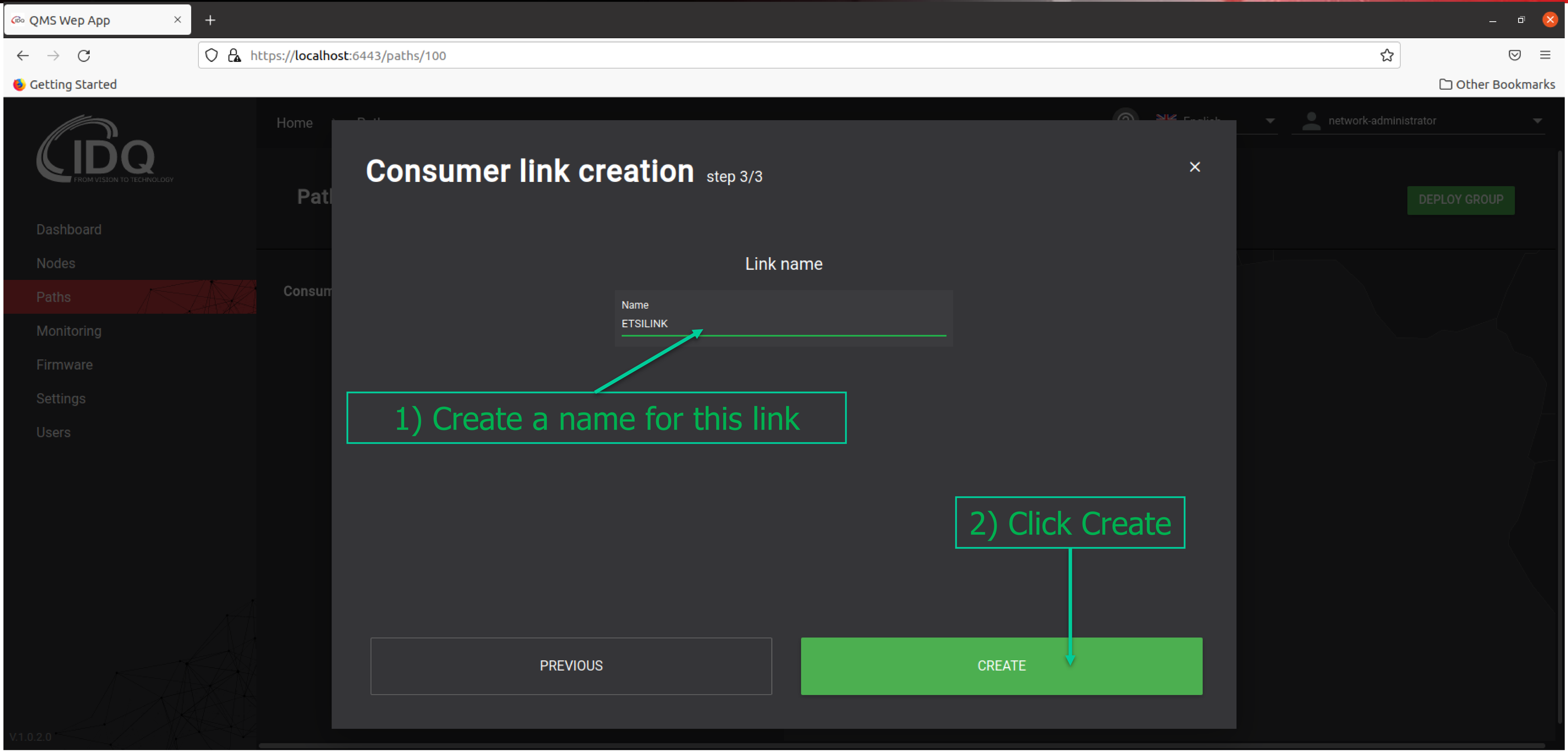
Create Consumer Links and Key Routing Path



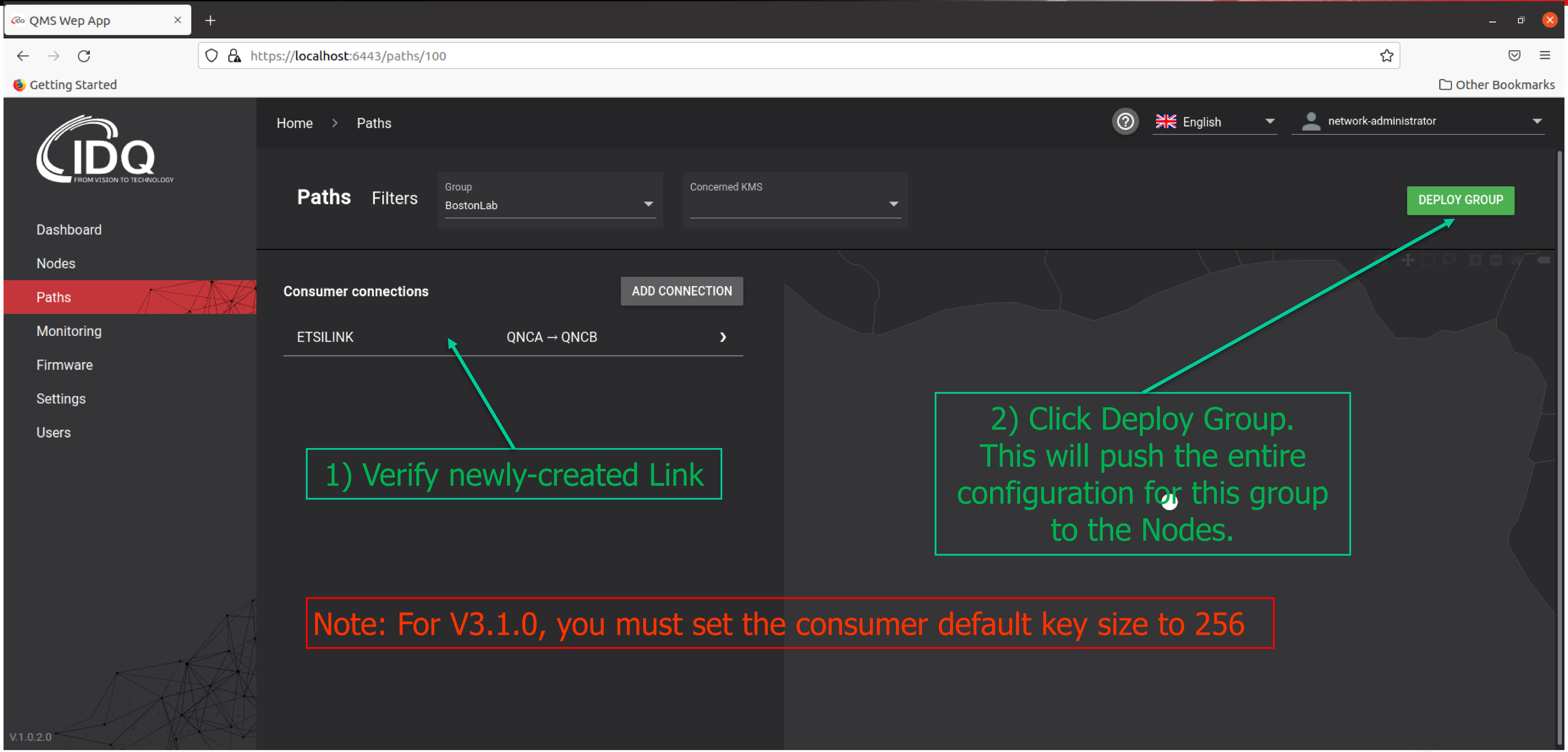
Create Consumer Links and Key Routing Path



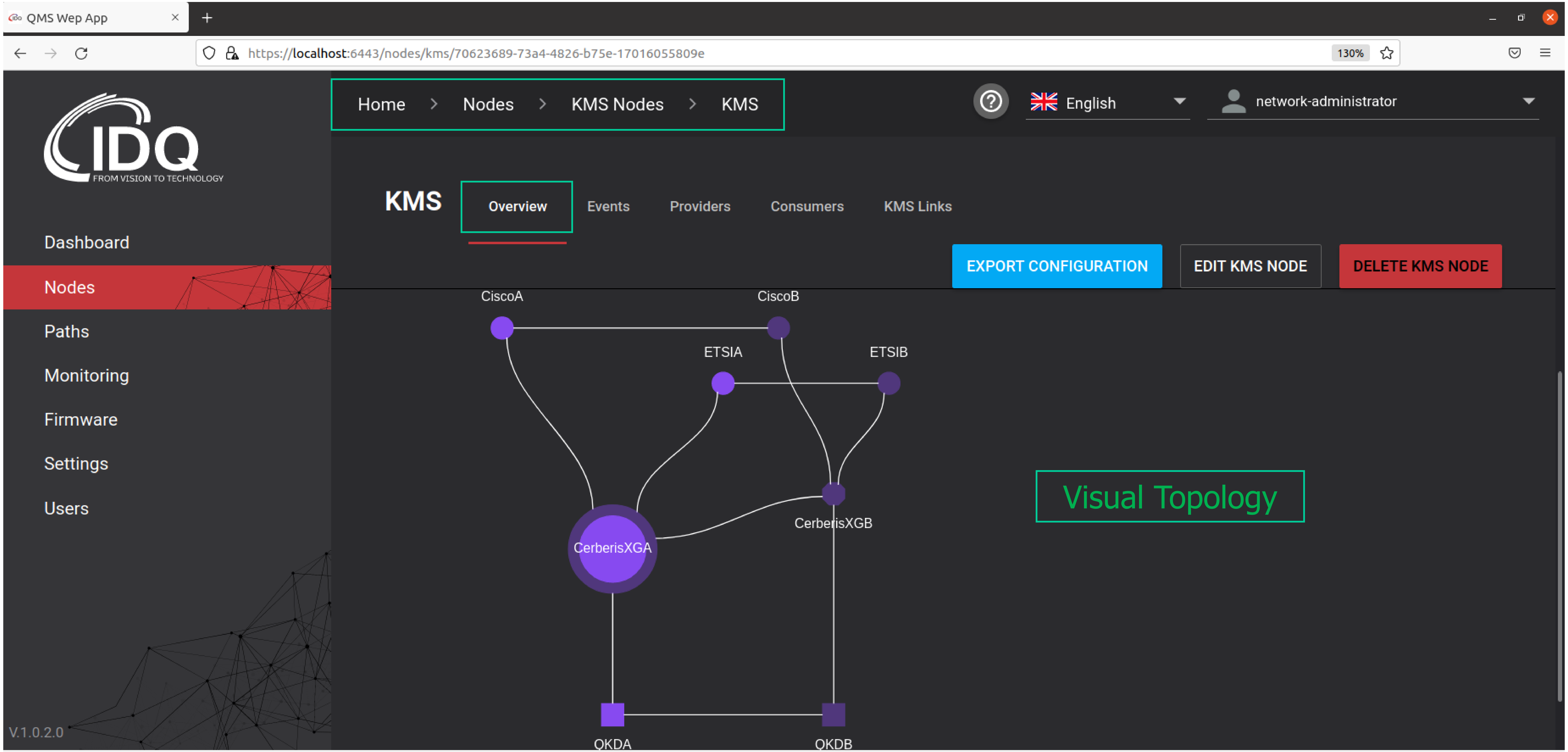
Create Consumer Links and Key Routing Path



Create Consumer Links and Key Routing Path



Verify Topology

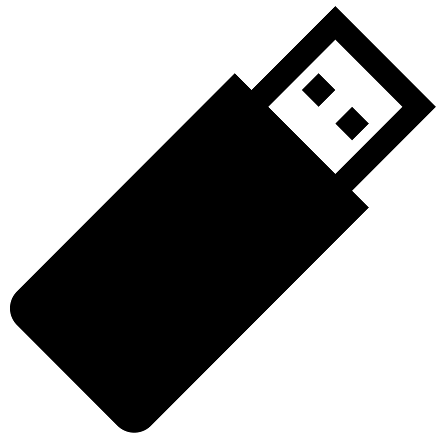


Certificate Configuration – Root CA

To set the correct certificates for the CerberisXG/QNC master and slave, we need to:

- **Add the certificate authority in CerberisXG/QNC**:
    - Plug the USB key with the certificates and keys into the CerberisXG or QNC USB port
    - Execute the command in CerberisXG/QNC CLI: ***encryptor add-client-ca*** then choose which client root CA to use

```
admin> encryptor add-client-ca
PEM files:
  1)   ChrisCA.pem
  2)   QKDServer.pem
  3)   QKDServer_pkey.pem
Please select a file (between 1 and 3).
```



Note: Use encryptor remove-client-ca command to remove unwanted CA files

Note: There are 4 USB ports on Cerberis3 QNC. Use the bottom/left USB port.

Certificate Configuration – Root CA

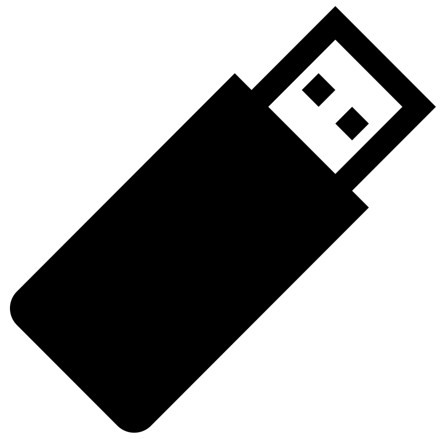
Repeat the same process to load the Certificate Authority on the peer QKD system

Certificate Configuration – Root CA

To set the correct certificates for the CerberisXG/QNC master and slave, we need to:

- **Add the certificate authority in CerberisXG/QNC**:
  - Plug the USB key with the certificates and keys into the CerberisXG or QNC USB port
  - Execute the command in CerberisXG/QNC CLI: ***encryptor add-client-ca*** then choose which client root CA to use

```
admin> encryptor add-client-ca
PEM files:
  1)   ChrisCA.pem
  2)   QKDServer.pem
  3)   QKDServer_pkey.pem
Please select a file (between 1 and 3).
```



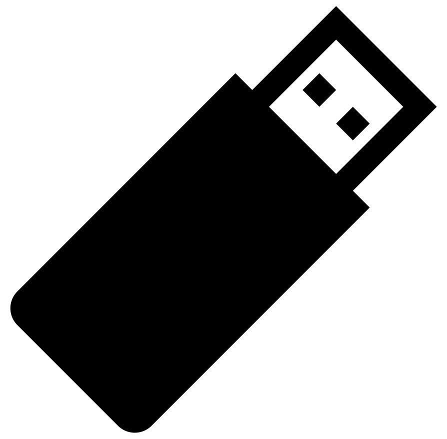
Note: Use encryptor remove-client-ca command to remove unwanted CA files

Note: There are 4 USB ports on Cerberis3 QNC. Use the bottom/left USB port.

Certificate Configuration – Server Certificates/Keys

- **Load the server Certificate and keys on CerberisXG/QNC**:
  - Plug the USB key with the certificates and keys into the CerberisXG/QNC USB port
  - Execute the command in CerberisXG/QNC CLI: ***encryptor load-key*** then copy server keys and server certificates on the system

```
admin> encryptor load-key --help
load key on the system

Usage: encryptor load-key [options]

Options:
  --help                  Show help information.
  -o|--overwrite          overwrite if already exists
  --cert-pwd <password>   certificate password

admin> encryptor load-key -o
PEM files:
  1)   ChrisCA.pem
  2)   QKDServer.pem
  3)   QKDServer_pkey.pem
Please select a file (between 1 and 3). 2
Key loaded
admin> encryptor load-key -o
PEM files:
  1)   ChrisCA.pem
  2)   QKDServer.pem
  3)   QKDServer_pkey.pem
Please select a file (between 1 and 3). 3
Key loaded
admin>
```



Load both the server cert and server key. You can use the –o option to overwrite existing key file.

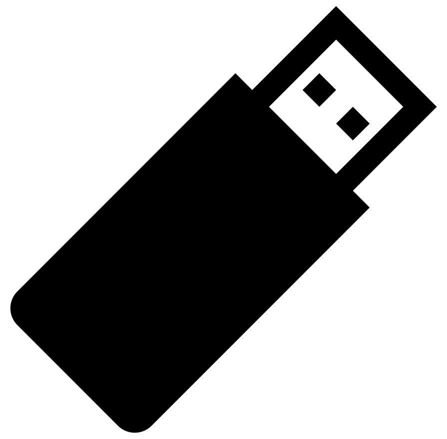
You'll need to enter the command "encryptor load-key" twice. Once for the certificate, and once for the key.

Repeat the same process to load the certificate and private key pair on the peer QKD system

Certificate Configuration – Server Certificates/Keys

- **Load the server Certificate and keys on CerberisXG/QNC**:
  - Plug the USB key with the certificates and keys into the CerberisXG/QNC USB port
  - Execute the command in CerberisXG/QNC CLI: ***encryptor load-key*** then copy server keys and server certificates on the system

```
admin> encryptor load-key --help
load key on the system

Usage: encryptor load-key [options]

Options:
  --help                  Show help information.
  -o|--overwrite          overwrite if already exists
  --cert-pwd <password>   certificate password

admin> encryptor load-key -o
PEM files:
  1)   ChrisCA.pem
  2)   QKDServer.pem
  3)   QKDServer_pkey.pem
Please select a file (between 1 and 3). 2
Key loaded
admin> encryptor load-key -o
PEM files:
  1)   ChrisCA.pem
  2)   QKDServer.pem
  3)   QKDServer_pkey.pem
Please select a file (between 1 and 3). 3
Key loaded
admin>
```



Load both the server cert and server key. You can use the –o option to overwrite existing key file.

You'll need to enter the command "encryptor load-key" twice. Once for the certificate, and once for the key.

Note: You may have a different certificate and key for Alice and Bob, depending on how you generated your certificates

Certificate Configuration – Assign Server Keys

SWISS QUANTUM

Set the server certificates and keys:

- Execute the command on the CerberisXG/QNC CLI: ***encryptor set-server-keys*** -> choose which server certificate and key to use to communicate with the encryptor

Note: File names must end in .pem extension

The consumers you created in QMS are displayed

```
admin> encryptor set-server-keys
        NAME       CERT-FILE             KEY-FILE           EXPIRATION                               SUBJECT
       ETSI1    kms-cert.pem           kms-key.pem  02/10/2023 04:05:00  CN=KME1, OU=QuantumSafe, O=ID Quantique, L=Gene...
   centaurisA  QKDServer.pem  QKDServer_pkey.pem  09/11/2035 14:42:05  CN=QKDServer, OU=QuantumSafe, O=ID Quantique, L...
Please type the consumer name for which you wish to change the server keys.
SYSTEM NAME:  centaurisA
CERT FILE:    QKDServer.pem
KEY FILE:      QKDServer_pkey.pem
Are you sure you want to continue? [Y/n]
```

Type the name of the consumer you want to set a new server cert/key for

Enter the new server cert and key that you imported in the previous screen

Use standalone command "encryptor" to verify configuration

IDQ

Certificate Configuration – Server Certificate/Key

Repeat the same process to assign the certificate and private key pair on the peer QKD system

Certificate Configuration – Assign Server Keys

Set the server certificates and keys:

- Execute the command on the CerberisXG/QNC CLI: ***encryptor set-server-keys*** -> choose which server certificate and key to use to communicate with the encryptor

Note: File names must end in .pem extension

The consumers you created in QMS are displayed

```
admin> encryptor set-server-keys
          NAME      CERT-FILE            KEY-FILE           EXPIRATION                                SUBJECT
         ETSI1   kms-cert.pem         kms-key.pem  02/10/2023 04:05:00  CN=KME1, OU=QuantumSafe, O=ID Quantique, L=Gene...
    centaurisA  QKDServer.pem  QKDServer_pkey.pem  09/11/2035 14:42:05  CN=QKDServer, OU=QuantumSafe, O=ID Quantique, L...
Please type the consumer name for which you wish to change the server keys.
SYSTEM NAME:  centaurisA
CERT FILE:    QKDServer.pem
KEY FILE:      QKDServer_pkey.pem
Are you sure you want to continue? [Y/n]
```

Type the name of the consumer you want to set a new server cert/key for

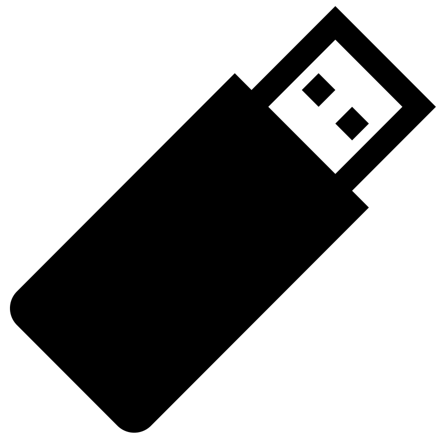
Enter the new server cert and key that you imported in the previous screen

Use standalone command "encryptor" to verify configuration

Add Initial Authentication Key

To create and synchronize the initial authentication key, plug a USB stick into the CerberisXG/master QNC (port 1 – bottom left USB port). Type the following command:

authkey generate



Follow instructions and then plug the USB stick to its peer CerberisXG/QNC. Connect to it and type:

authkey load

You can verify the successful auth key load using the logtail pagti command.
If an error occurs during creation or loading of the auth key, restart the KMS service and retry. (restart –help)

Verify ISK has been set

SWISS QUANTUM

View the QKD logs:
On the QKD you should be able to see "set initial key".
The system will not start generating quantum keys without it.

If you don't see this message, verify the date/time is properly set on each node/module and QMS. Verify there is no NAT on your KMS->KMS communication

```
2020-02-19T16:36:55.374 DEBUG [(anonymous namespace)::FsmFrontend::NormalOperation_::ExecutingGeneralInitialization_*] Leaving
 ExecutingGeneralInitialization state.
2020-02-19T16:36:55.375 DEBUG [(anonymous namespace)::FsmFrontend::NormalOperation_::ExecutingSecurityInitialization*] Enterin
g ExecutingSecurityInitialization state.
2020-02-19T16:40:52.137 INFO  [cervino::communicator::idq4p::CommandDispatcher] Received IDQ4P command 'GetSystemState'.
2020-02-19T16:42:20.609 INFO  [cervino::communicator::idq4p::CommandDispatcher] Received IDQ4P command 'SetInitialKey'.
2020-02-19T16:42:20.609 DEBUG [(anonymous namespace)::FsmFrontend*] Processing event SetInitialKey.
2020-02-19T16:42:20.610 INFO  [cervino::communicator::backend::KeyExchangeBob] Setting initial authentication key...
2020-02-19T16:42:20.768 INFO  [cervino::communicator::backend::KeyExchangeBob] Initial authentication key set.
2020-02-19T16:42:20.768 DEBUG [(anonymous namespace)::FsmFrontend*] Processing event StartKeyExchange.
2020-02-19T16:42:20.768 DEBUG [(anonymous namespace)::FsmFrontend::NormalOperation_::ExecutingSecurityInitialization*] Leaving
 ExecutingSecurityInitialization state.
2020-02-19T16:42:20.768 DEBUG [(anonymous namespace)::FsmFrontend::NormalOperation_::Running*] Entering Running state.
```


IDQ

Key Generation

SWISS QUANTUM

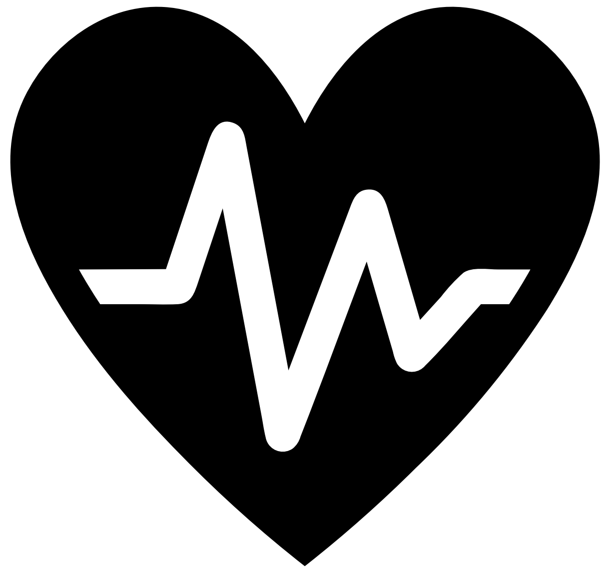
```
2020-02-19T16:46:20.808 DEBUG [cervino::communicator::backend::KeyExchangeBob] Actual key rate: 0 bits/s
2020-02-19T16:46:30.806 DEBUG [cervino::communicator::backend::KeyExchangeBob] Actual key rate: 0 bits/s
2020-02-19T16:46:31.796 INFO  [cervino::communicator::backend::OpticsRegulationBob] Regulating detection ratio: ratio = 1.015
5, DATA delay = 2398 ps, MONITOR delay = 2189 ps.
2020-02-19T16:46:40.810 DEBUG [cervino::communicator::backend::KeyExchangeBob] Actual key rate: 0 bits/s
2020-02-19T16:46:42.808 INFO  [cervino::communicator::backend::OpticsRegulationBob] Regulating detection ratio: ratio = 1.134
6, DATA delay = 2409 ps, MONITOR delay = 2200 ps.
2020-02-19T16:46:50.813 DEBUG [cervino::communicator::backend::KeyExchangeBob] Actual key rate: 947.2 bits/s
2020-02-19T16:46:53.811 INFO  [cervino::communicator::backend::OpticsRegulationBob] Regulating detection ratio: ratio = 1.375
5, DATA delay = 2420 ps, MONITOR delay = 2211 ps.
2020-02-19T16:47:00.813 DEBUG [cervino::communicator::backend::KeyExchangeBob] Actual key rate: 2073.6 bits/s
2020-02-19T16:47:04.817 INFO  [cervino::communicator::backend::OpticsRegulationBob] Regulating detection ratio: ratio = 1.165
6, DATA delay = 2409 ps, MONITOR delay = 2200 ps.
2020-02-19T16:47:10.814 DEBUG [cervino::communicator::backend::KeyExchangeBob] Actual key rate: 2150.4 bits/s
2020-02-19T16:47:15.825 INFO  [cervino::communicator::backend::OpticsRegulationBob] Regulating detection ratio: ratio = 1.036
1, DATA delay = 2398 ps, MONITOR delay = 2189 ps.
2020-02-19T16:47:20.808 DEBUG [cervino::communicator::backend::KeyExchangeBob] Actual key rate: 2150.4 bits/s
2020-02-19T16:47:26.828 INFO  [cervino::communicator::backend::OpticsRegulationBob] Regulating detection ratio: ratio = 1.149
```

After few minutes you should see key exchange. Check Alice logs to validate proper QBER, Visibility and Key Block Filling Rate

Monitoring and Statistics

SWISS QUANTUM

Monitor vital system statistics



Monitoring/Statistics Preparation

Verify if monitoring is enabled on each QKD systems using QNET:

In your QNET terminal, type: “qnet monitor node <node_name>”

Verify clid and idq4p options enabled

```
chris@chris-VirtualBox:~/Documents$ qnet monitor node CerberisXGA
KMS node CerberisXGA monitoring options: cpu ram temperature cooling-fan power-48 clid idq4p log:Info
chris@chris-VirtualBox:~/Documents$ qnet monitor node CerberisXGB
KMS node CerberisXGB monitoring options: cpu ram temperature cooling-fan power-48 clid idq4p log:Info
chris@chris-VirtualBox:~/Documents$
```

See QNET Tool User Guide for complete list of monitoring options

Monitoring/Statistics Preparation

Enable monitoring on both QKD systems if it is not enabled:

In your QNET terminal, type: “qnet monitor node <node_name> -clid -idq4p”

Once for each QKD system

```
chris@chris-VirtualBox:/home/idq/qnet-package-qms-1.1.6$ qnet monitor node CerberisXGA -clid -idq4p
KMS node CerberisXGA monitoring configuration successfully updated.
chris@chris-VirtualBox:/home/idq/qnet-package-qms-1.1.6$ qnet monitor node CerberisXGB -clid -idq4p
KMS node CerberisXGB monitoring configuration successfully updated.
chris@chris-VirtualBox:/home/idq/qnet-package-qms-1.1.6$
```

Monitoring/Statistics Preparation

Now the QNC webAPI must be restarted on each node, to do that one of the following methods can be used:

1. Via SSH connection:
   Log-in as admin in the QNET Shell of each node:

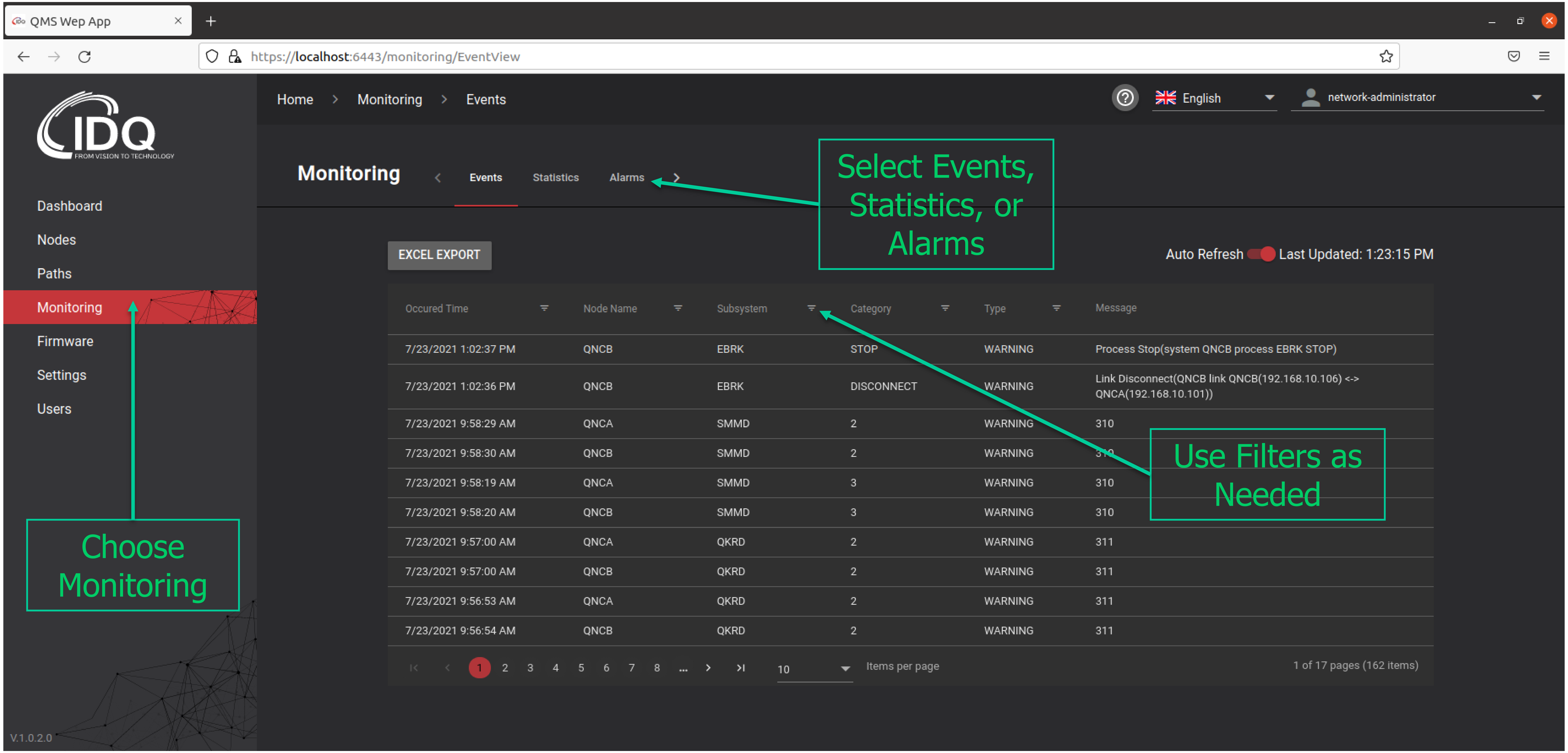
```
ssh admin@10.10.10.51
   admin> restart webapi
   QNC WebApi service will restart.
   Are you sure you want to continue? [Y/n] Y
   Service adminwebapi restarted

   admin> restart kms
   The key management service will restart.
   Are you sure you want to continue? [Y/n] Y
   Service key management restarted
```

```
ssh admin@10.10.10.52
   admin> restart webapi
   QNC WebApi service will restart.
   Are you sure you want to continue? [Y/n] Y
   Service adminwebapi restarted

   admin> restart kms
   The key management service will restart.
   Are you sure you want to continue? [Y/n] Y
   Service key management restarted
```

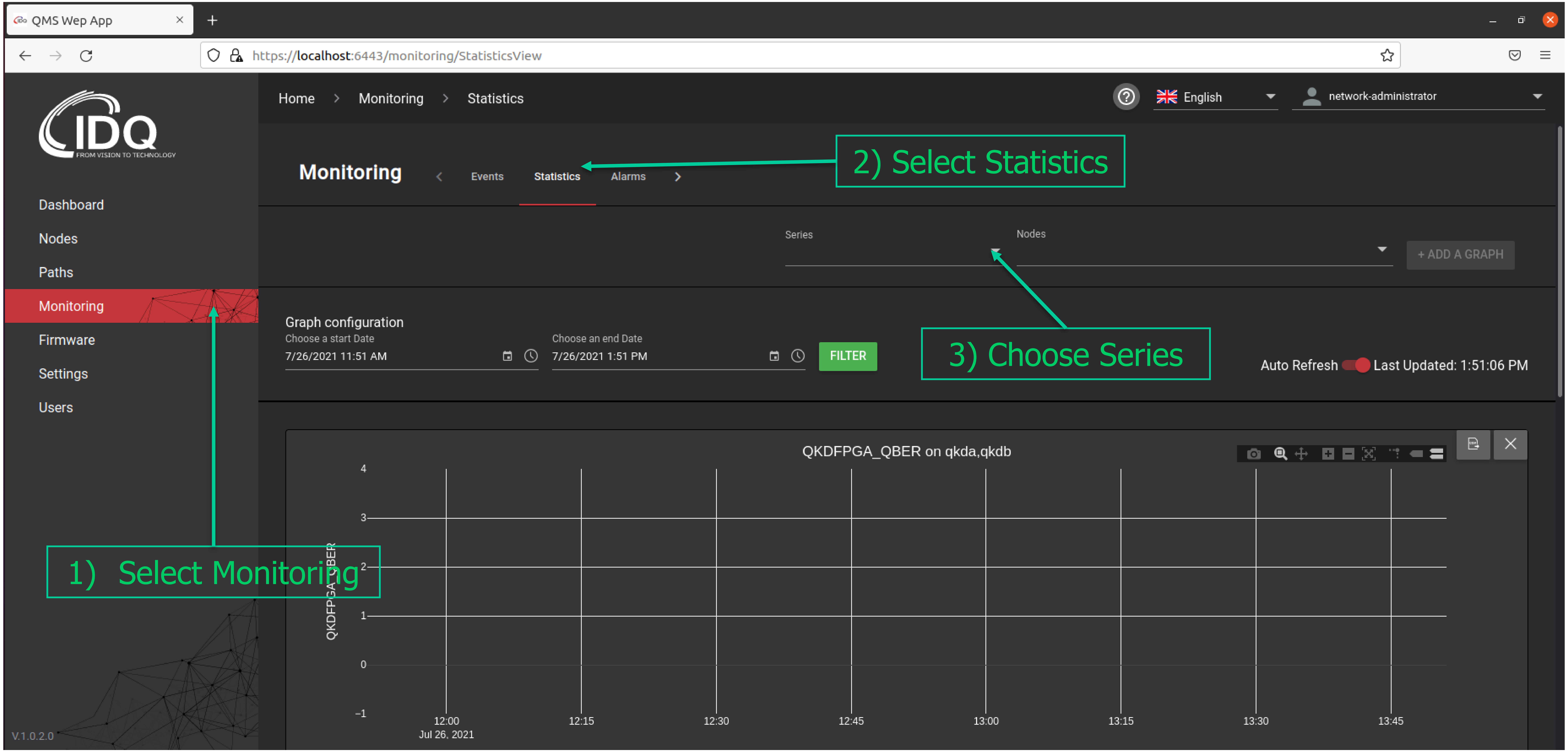
2. With QNET Tools:

```
> qnet reboot node CerberisXGA
> qnet reboot node CerberisXGB
```

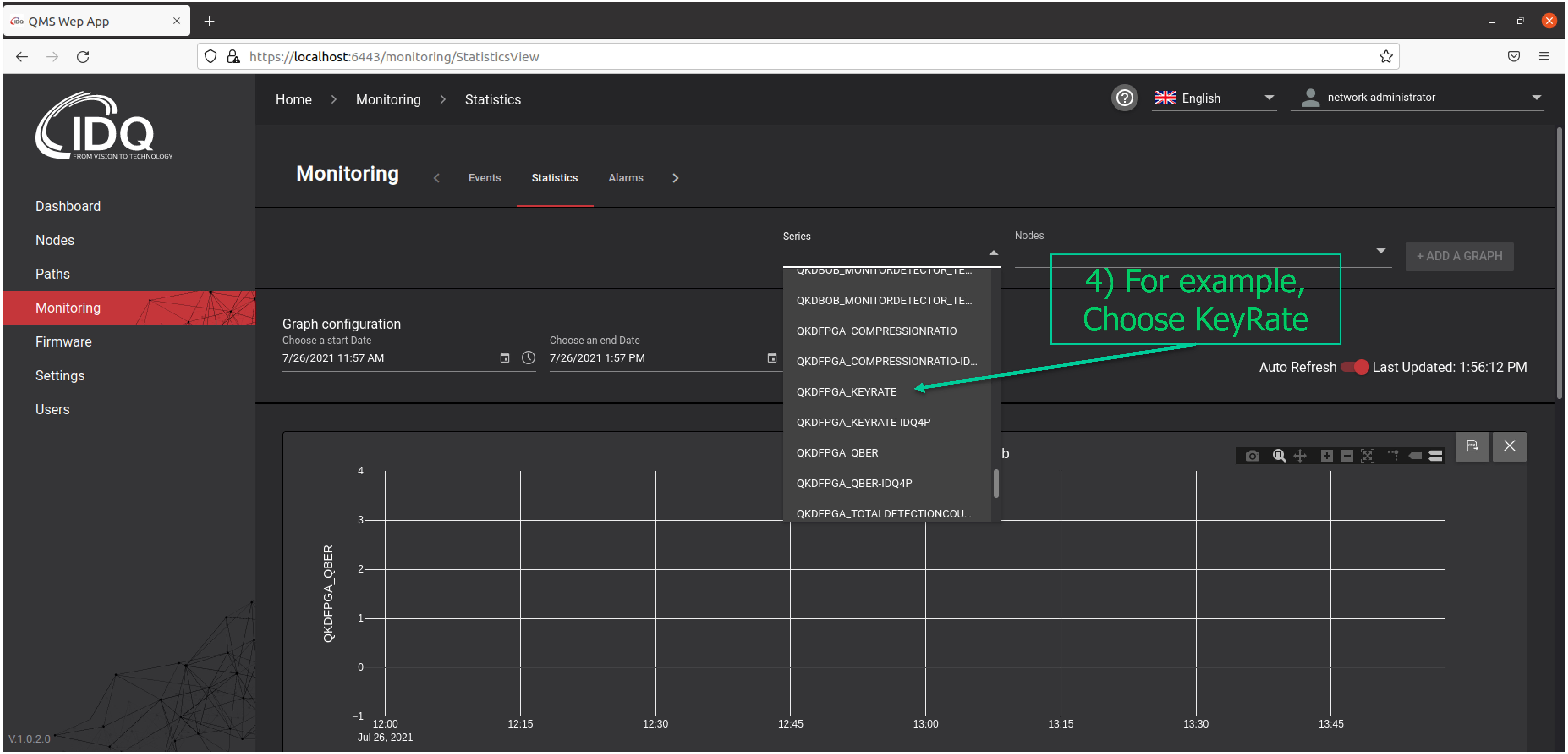
Monitoring/Statistics



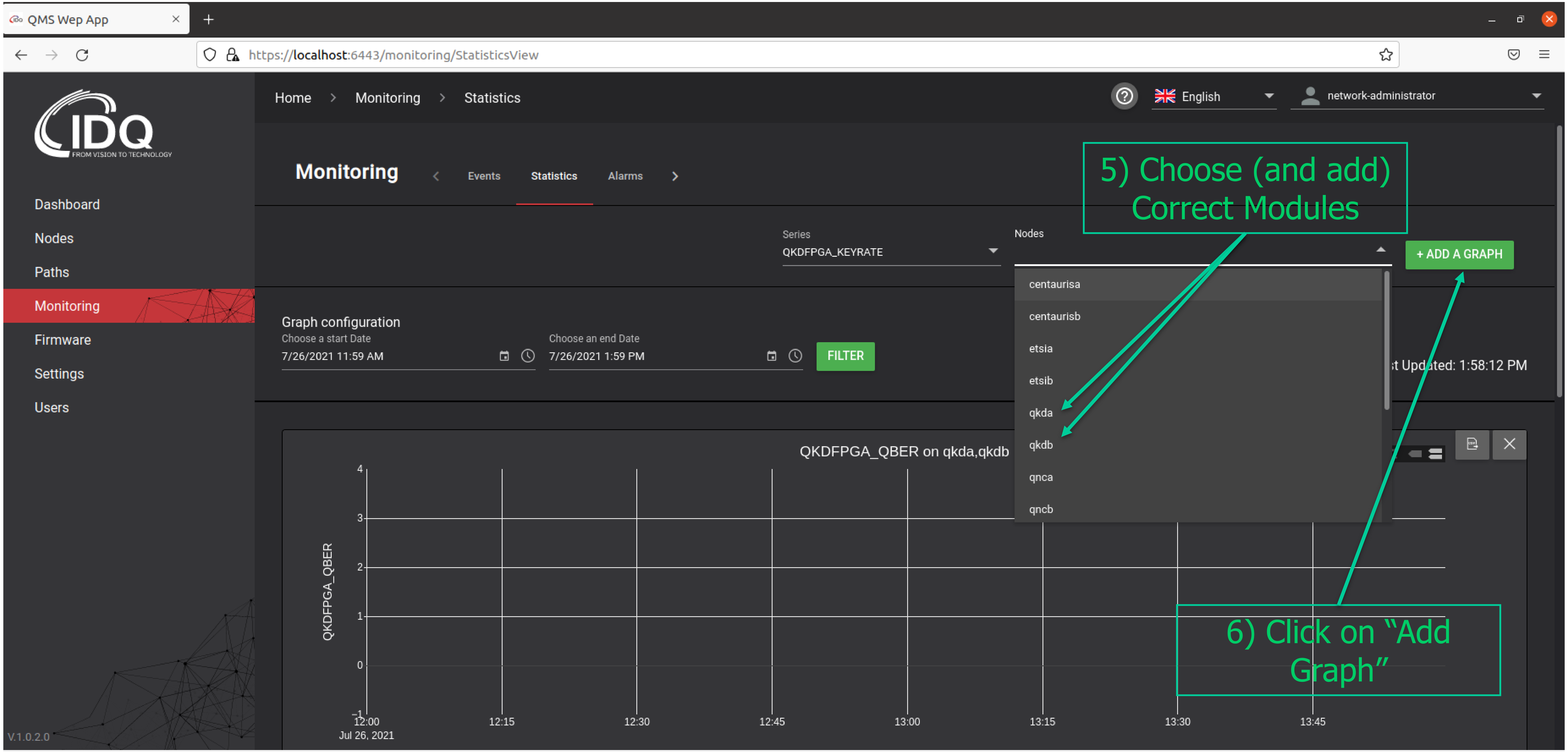
Generating Custom Graphs/Statistics



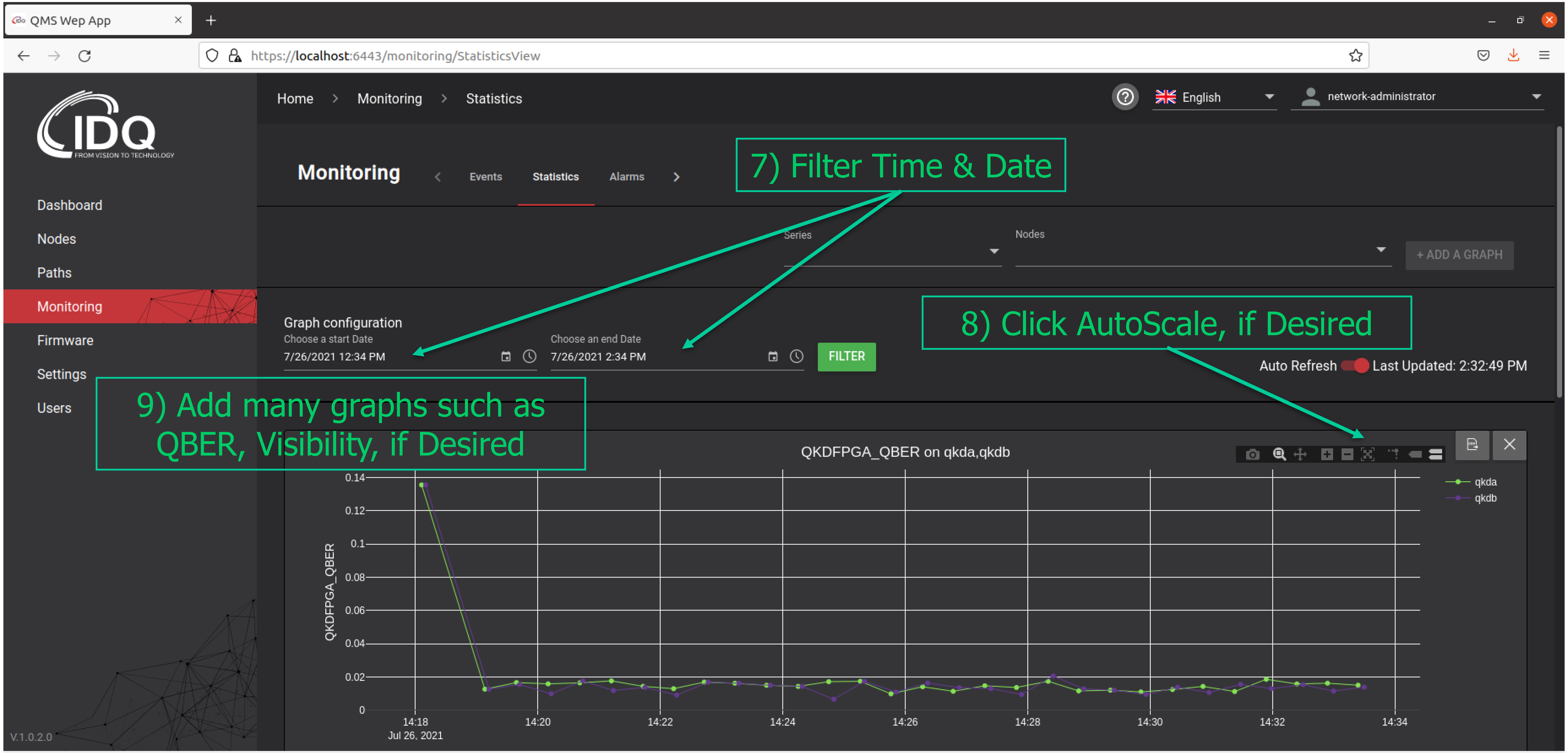
Generating Custom Graphs/Statistics



Generating Custom Graphs/Statistics

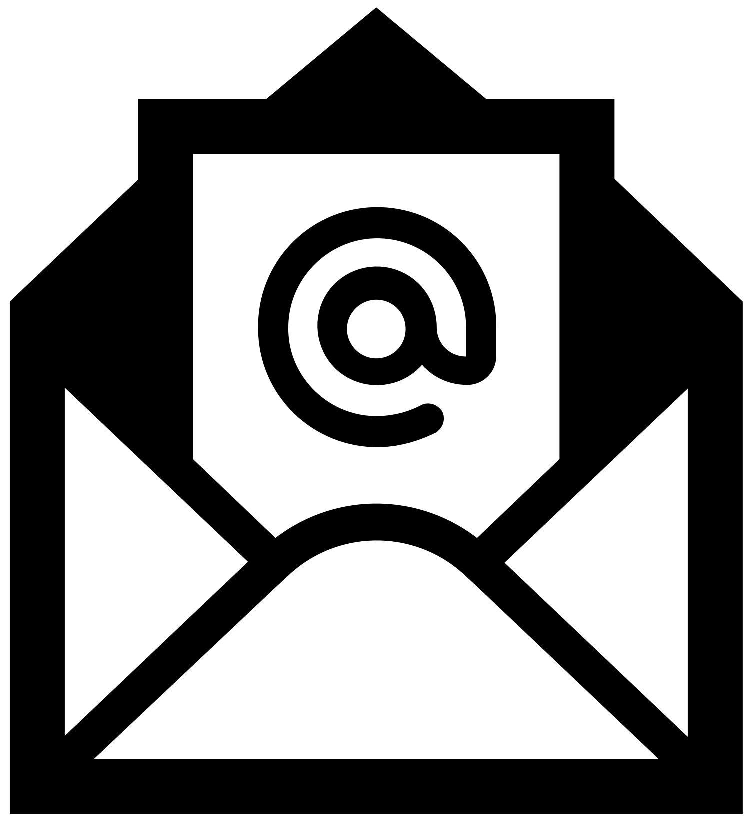


Generating Custom Graphs/Statistics



Email Notifications – Event & Alarm Triggers

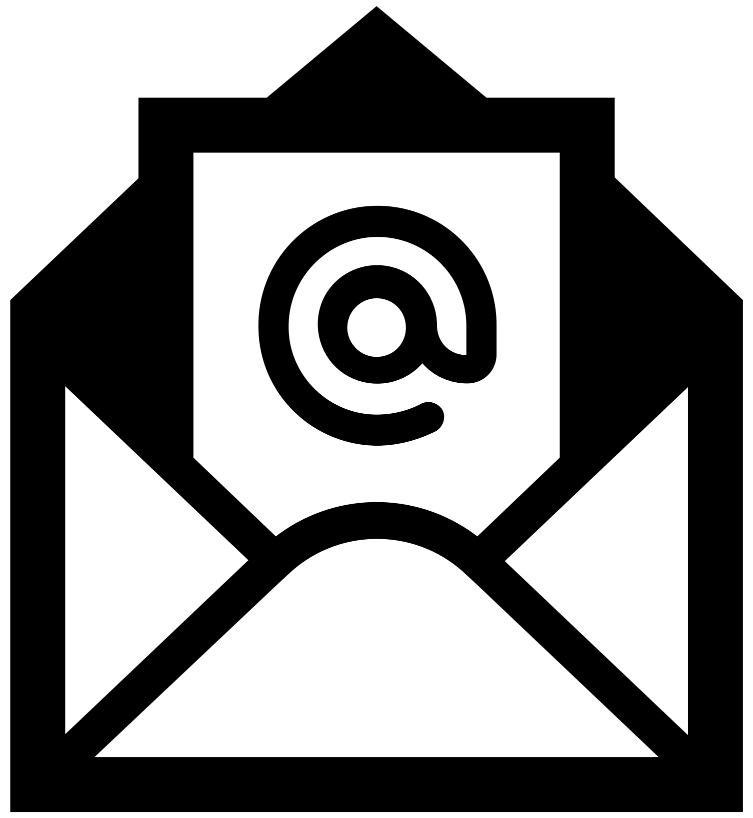
- To create custom notifications based on events/triggers, we must first:
  - Create a Notification Template
  - Create Email Distribution Lists
  - Create Triggers
  - Modify QMS docker-compose.override.yml file with SMTP Information



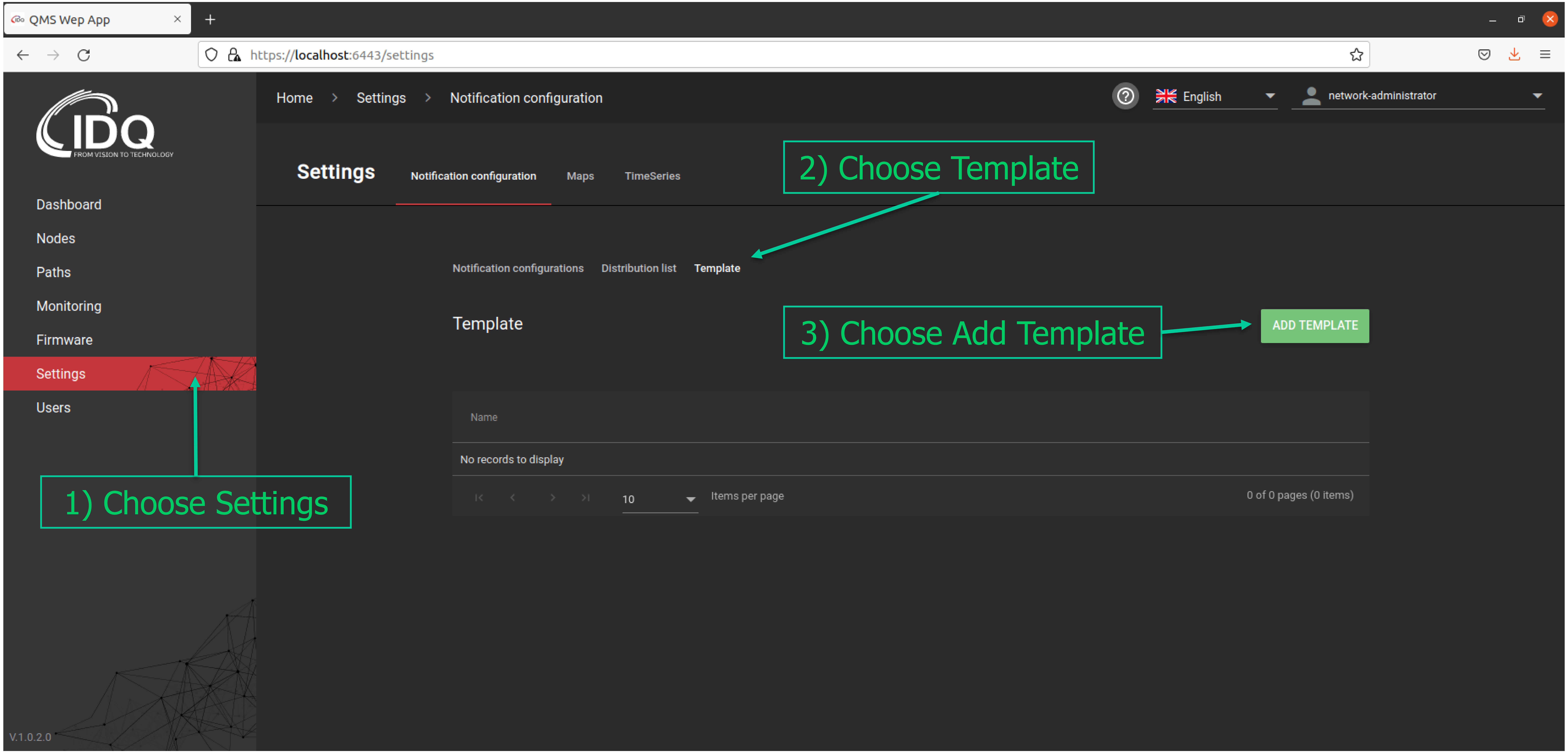
Email Notifications – Event & Alarm Triggers

SWISS QUANTUM

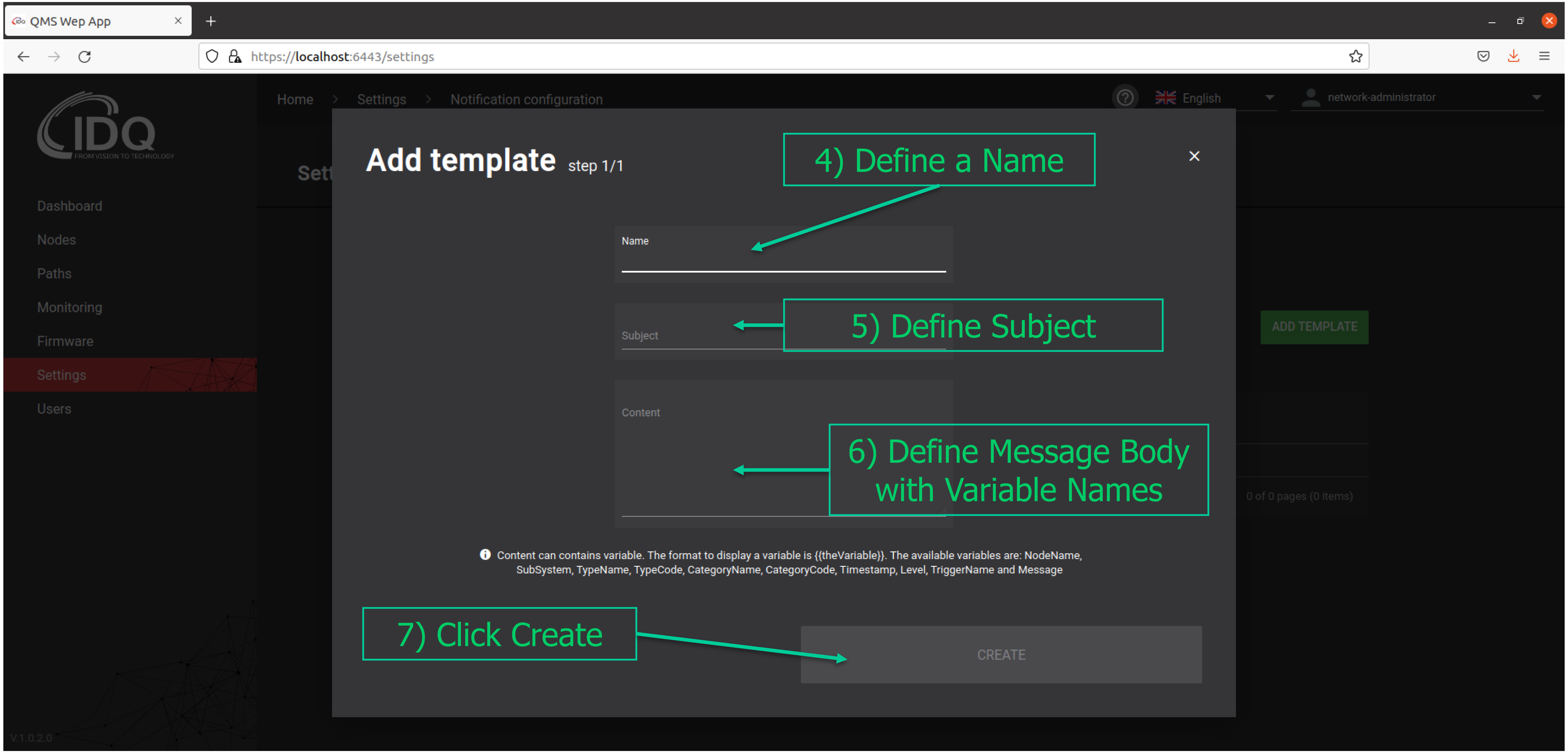
Add a Template



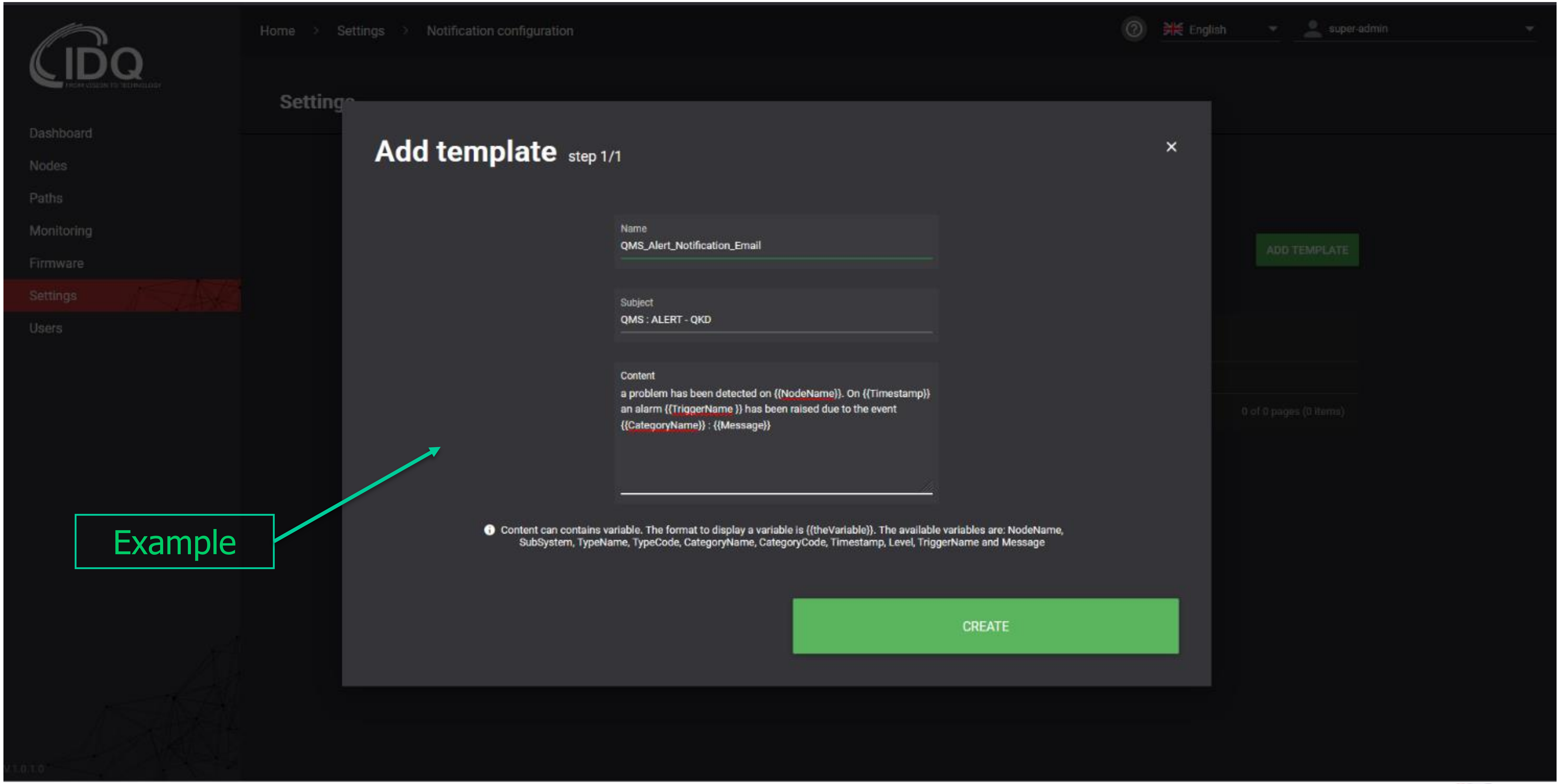
Email Notifications – Event & Alarm Triggers



Email Notifications – Event & Alarm Triggers

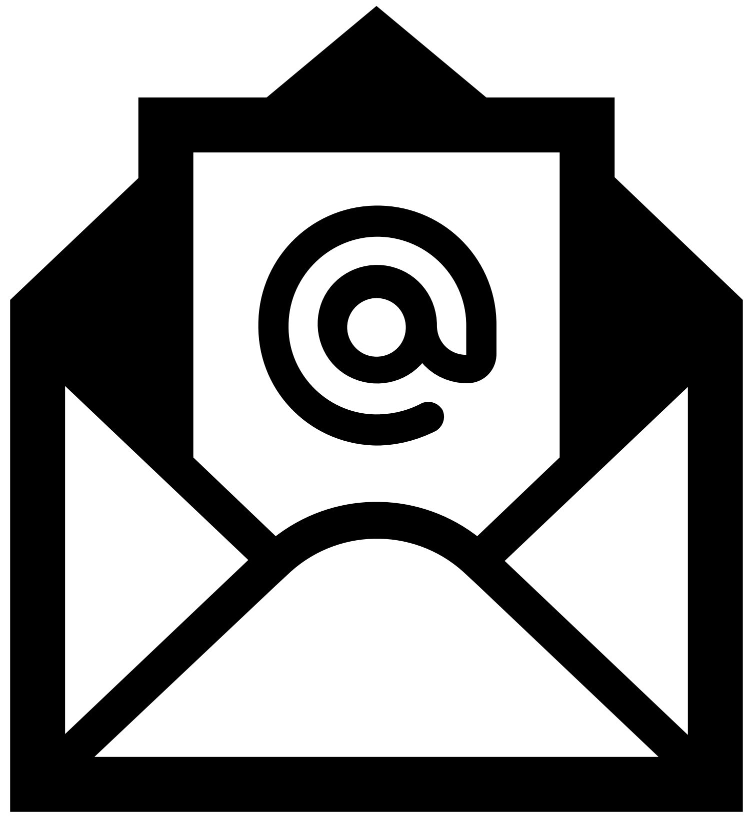


Email Notifications – Event & Alarm Triggers

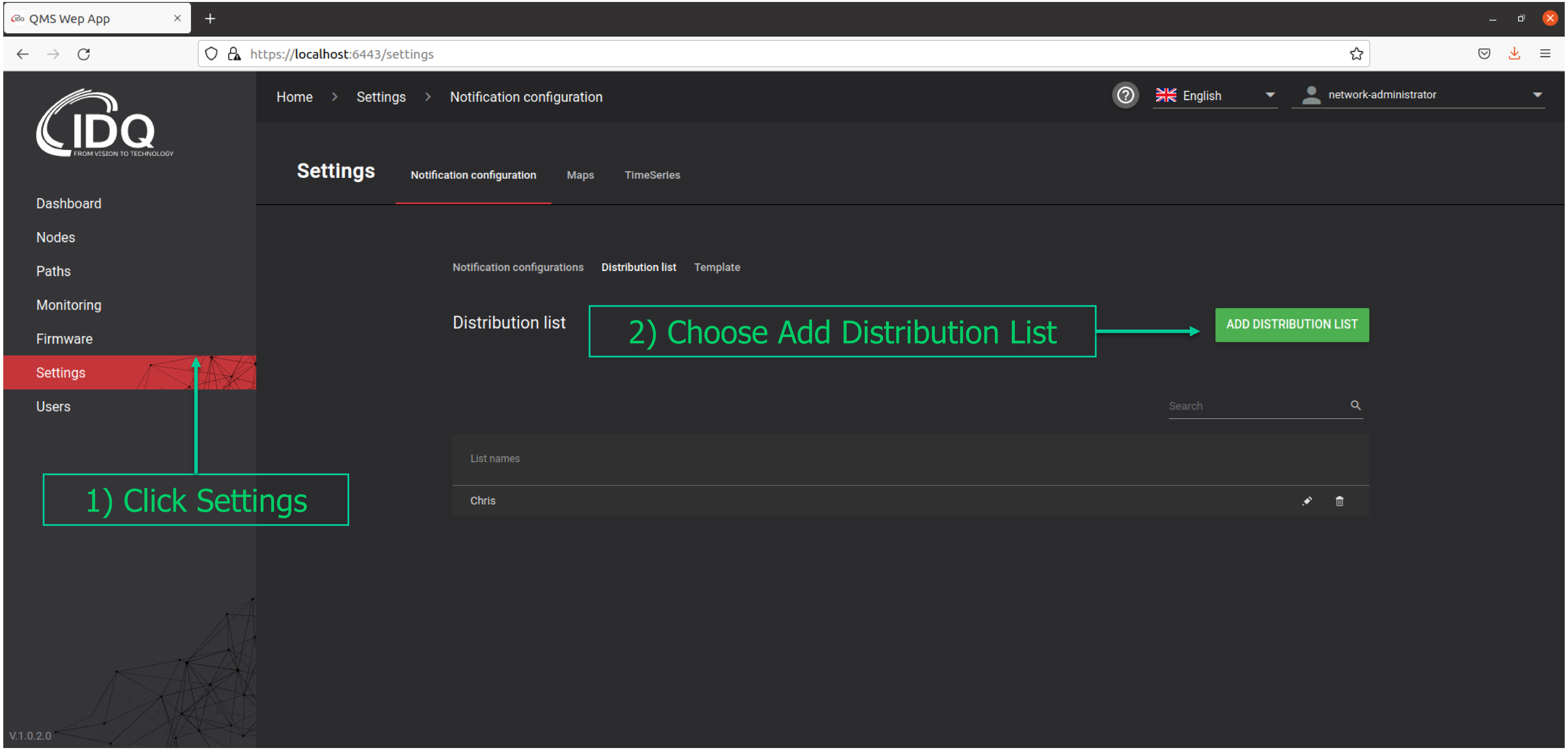


SWISS QUANTUM

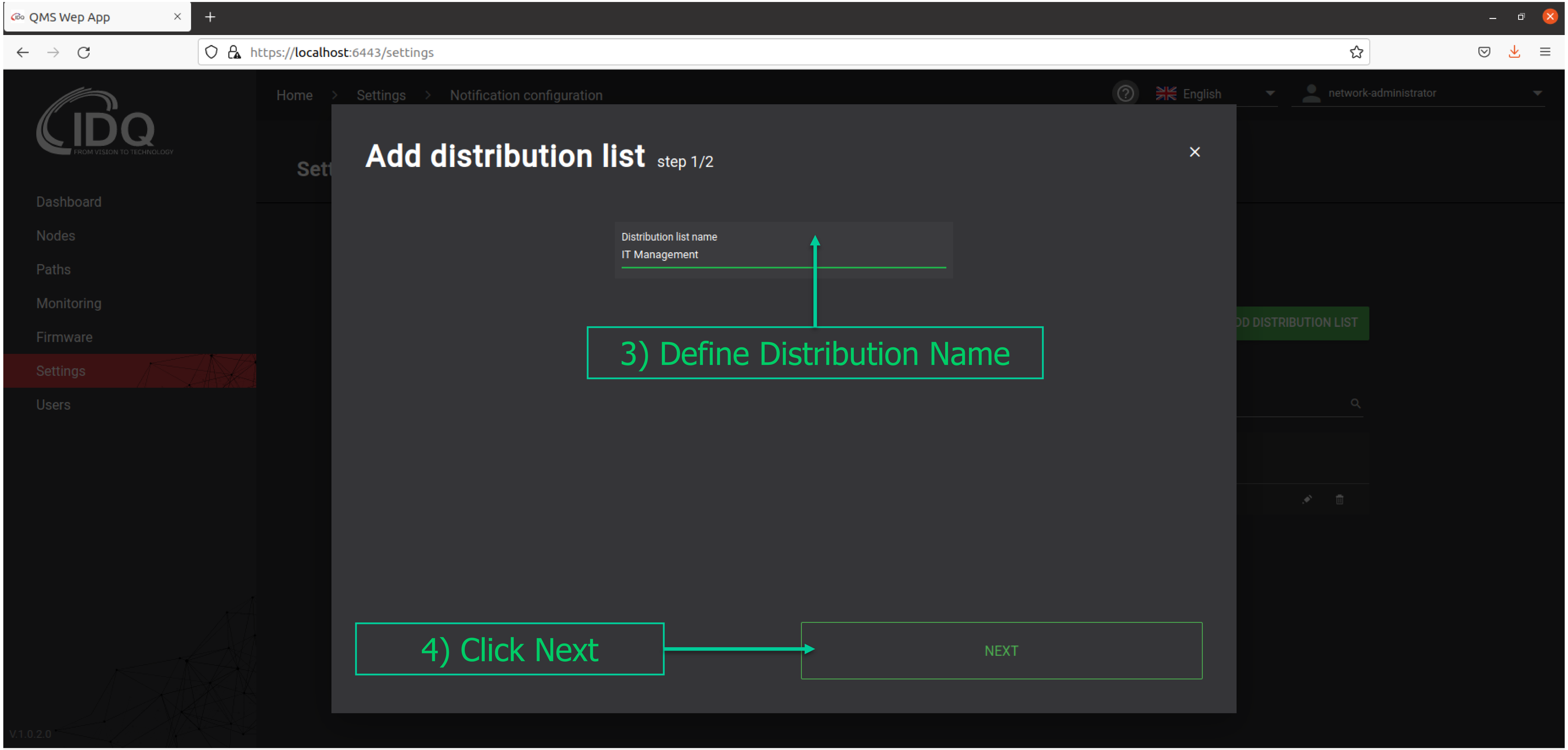
Create Distribution List



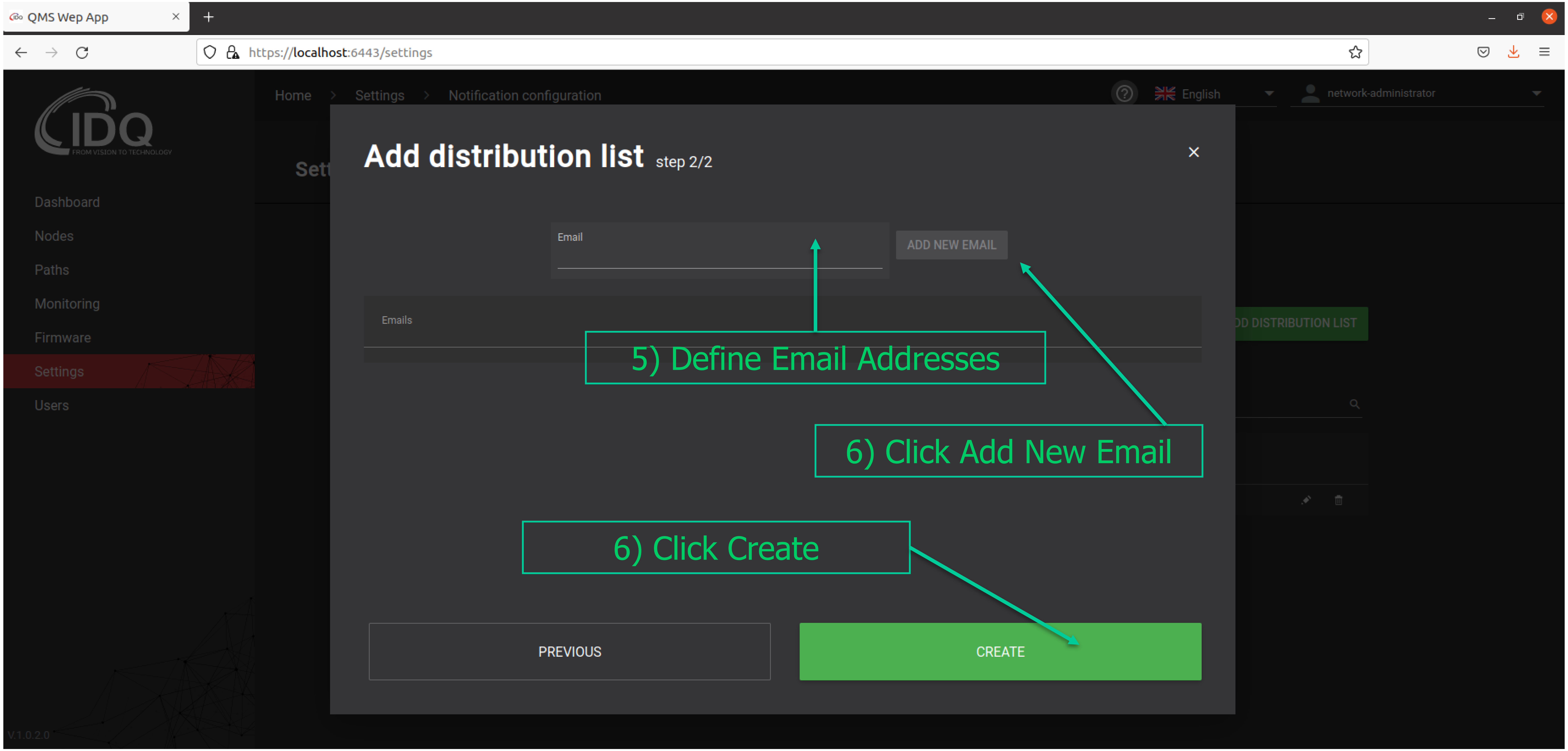
Email Notifications – Event & Alarm Triggers



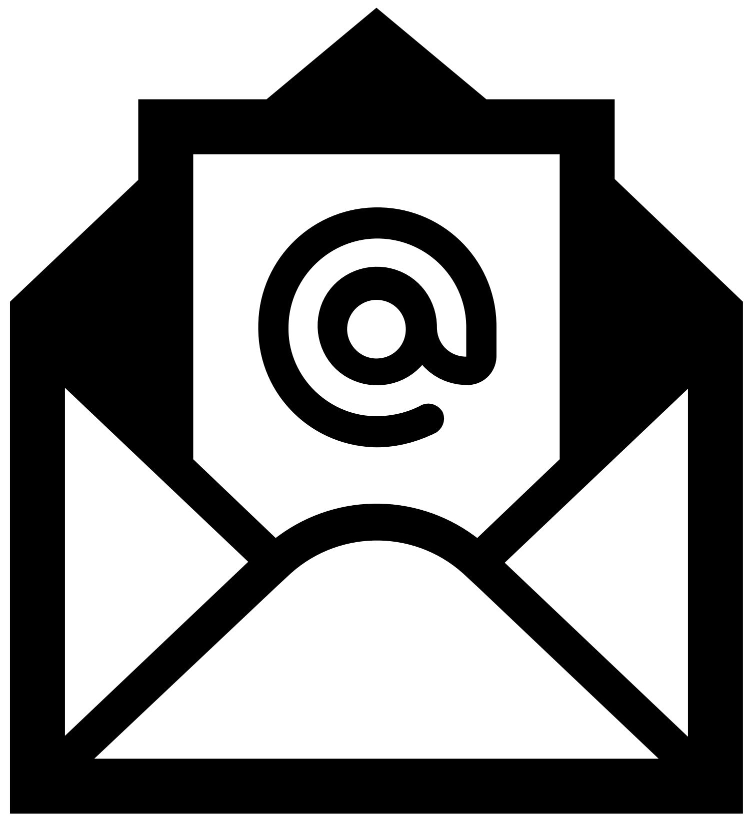
Email Notifications – Event & Alarm Triggers



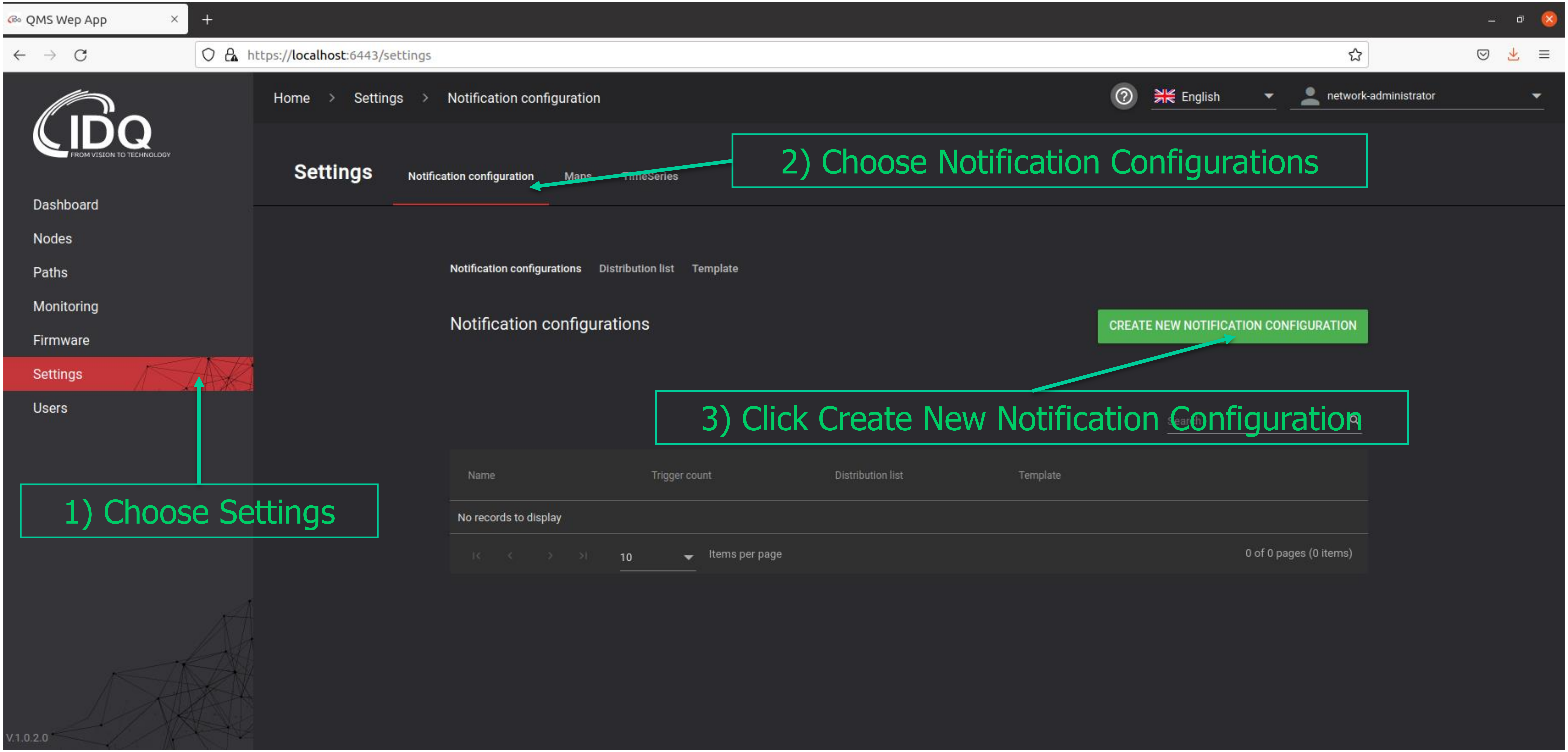
Email Notifications – Event & Alarm Triggers



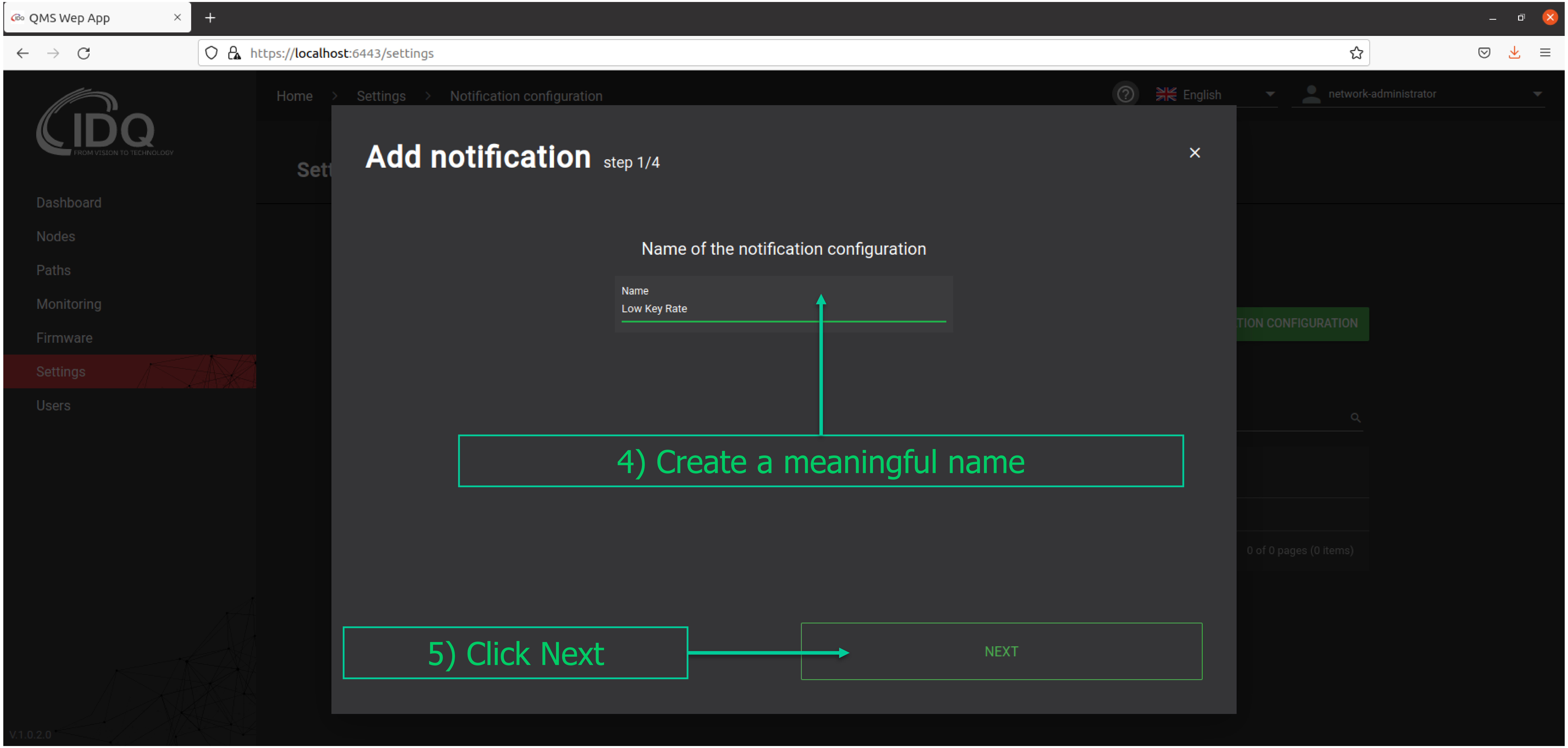
Add a Notification Trigger



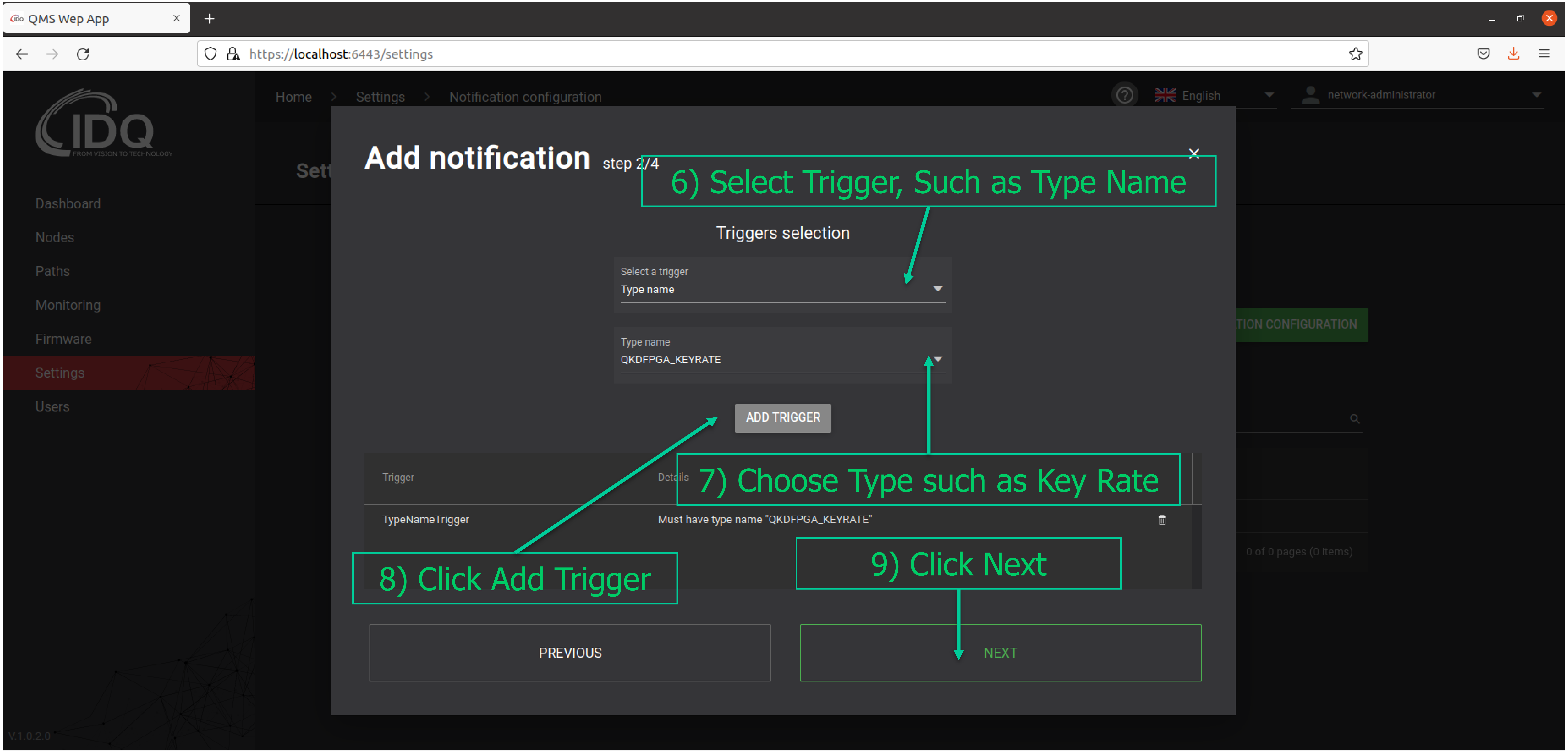
Email Notifications – Event & Alarm Triggers



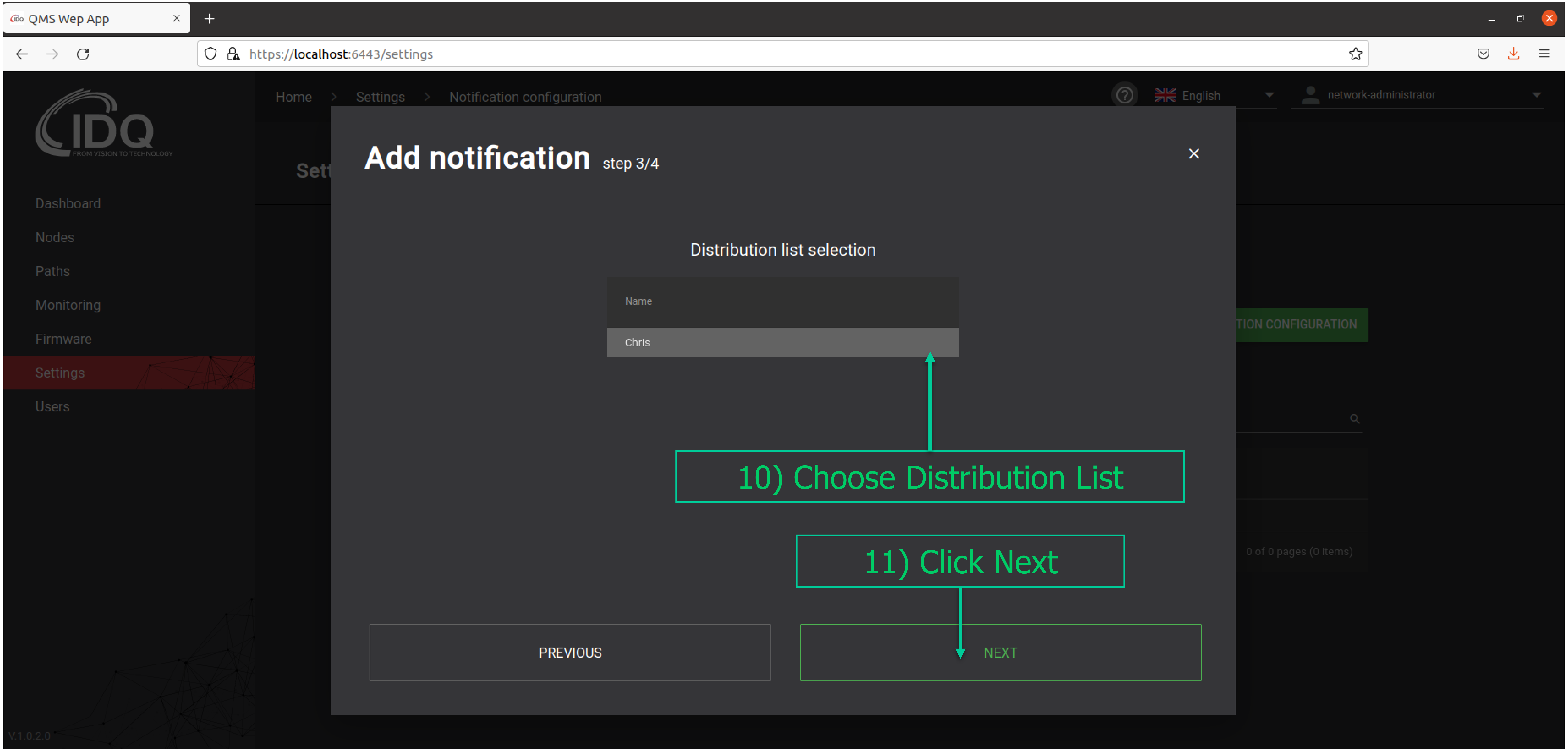
Email Notifications – Event & Alarm Triggers



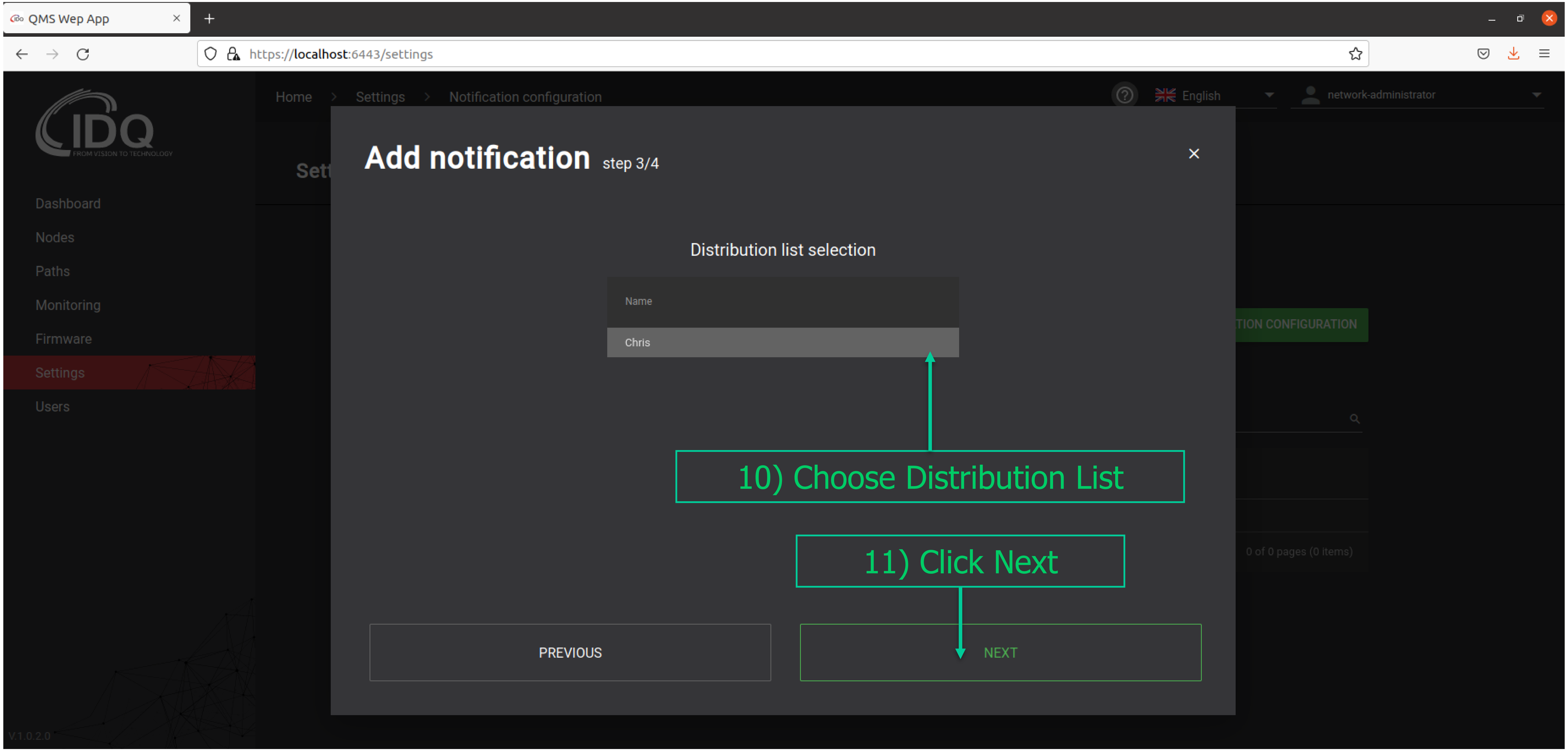
Email Notifications – Event & Alarm Triggers



Email Notifications – Event & Alarm Triggers

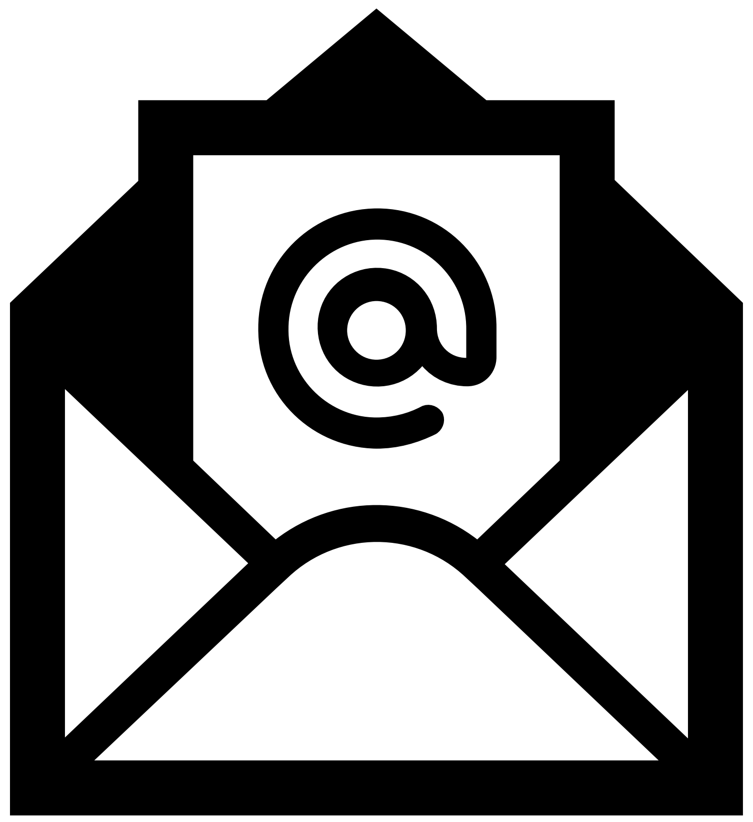


Email Notifications – Event & Alarm Triggers



Email Notifications – Event & Alarm Triggers

Modify docker-compose.override.yml



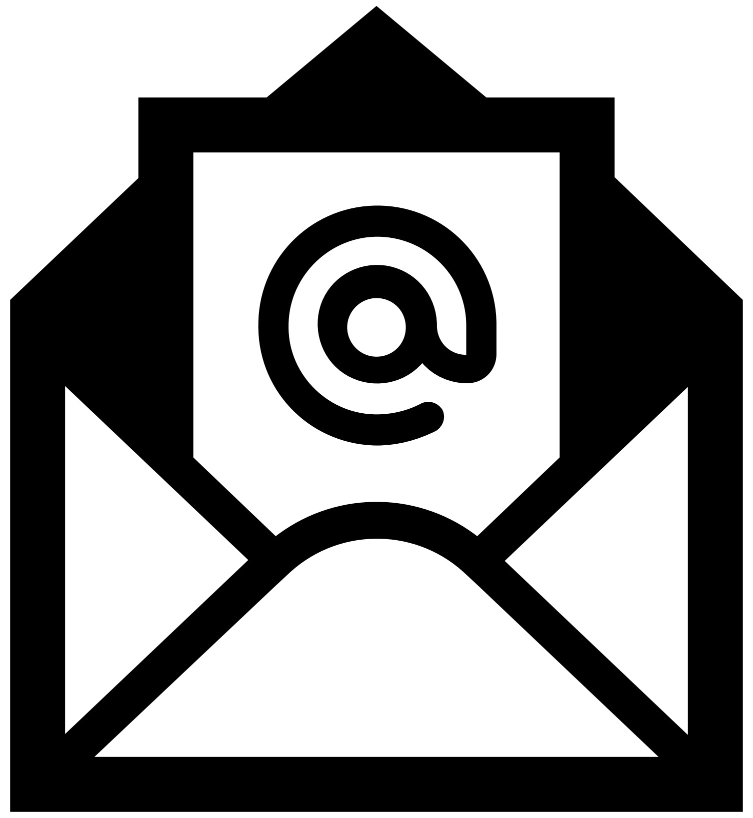
Email Notifications – Event & Alarm Triggers

SWISS QUANTUM

Configure SMTP

To enable QMS to send notifications by email, you need an existing SMTP server reachable by the QNET central server. Add the required parameters (credentials, sender info, etc.) as indicated in the `docker-compose.override.yml` file.

The next slide shows an example of a SMTP configuration in the `docker-compose.override.yml` file:



Email Notifications – Event & Alarm Triggers

```yaml
qms-backend:
  # Configuration of SMTP server for sending alarm notification by email
  environment:
    # Hostname and port of the SMTP server
    SMTP_SERVER: "smtp.mycompany.com"
    SMTP_PORT: 1025
    # Credentials to use for connection to SMTP server
    SMTP_USERNAME: "qms"
    SMTP_PASSWORD: "1234ç%&$_AbcDef"
    # Emails sent to distribution lists will have this sender
    SMTP_SENDER_NAME: "QMS notifications"
    SMTP_SENDER_ADDRESS: "qms-notifications@mycompany.com"
```

Email Notifications – Event & Alarm Triggers

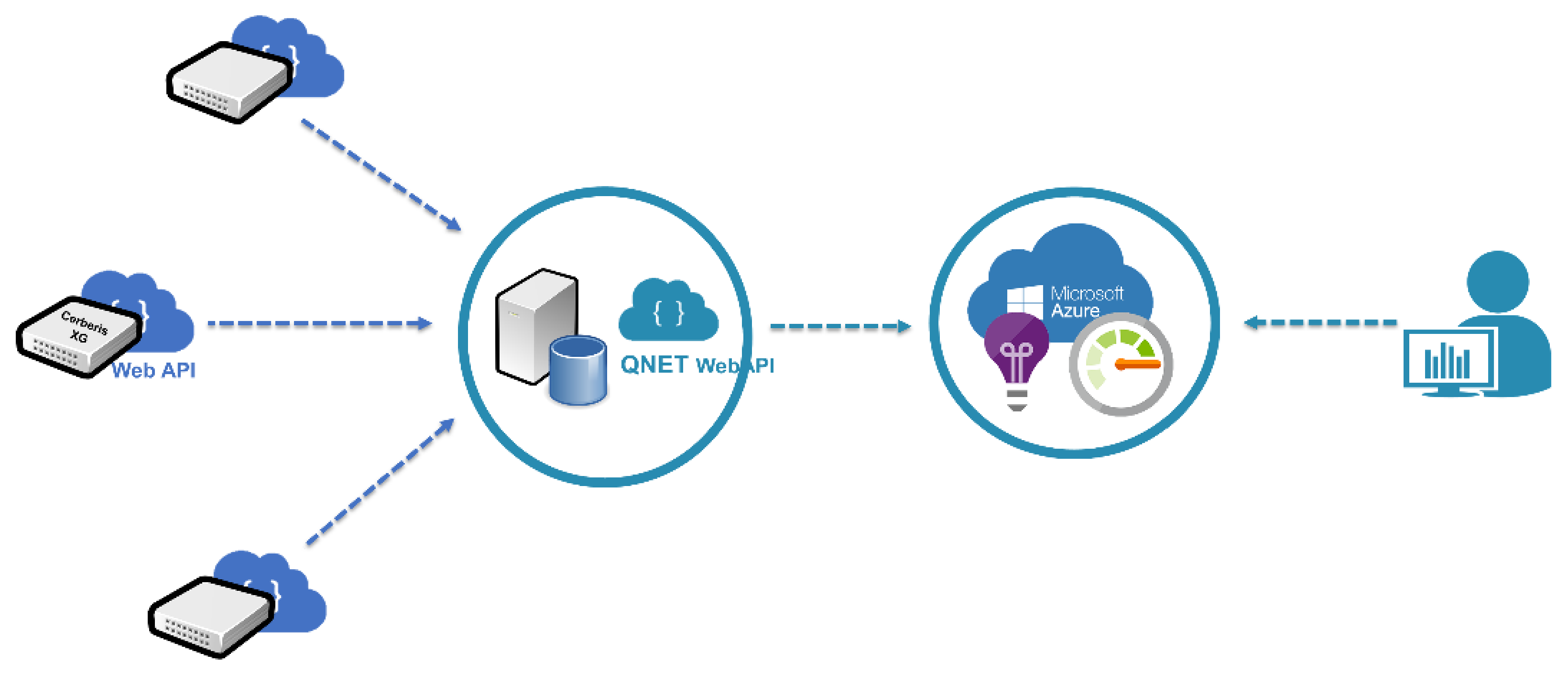
Config exists in-between qms-frontend and qms-postgres

Will need to stop and restart the docker container:
(This will terminate your browswer session)

sudo docker-compose down
sudo docker-compose up -d

```yaml
qms-frontend:
  # Certificates for SSL/TLS
  volumes:
    - ./volumes/certs:/certs:ro
  environment:
    # Certificates for SSL/TLS
    # RSA Private key for SSL/TLS configuration
    - TLS_KEY_FILE_PATH=/certs/qms.key
    # Certificate with CA-chain for SSL/TLS configuration
    - TLS_CERT_AND_CHAIN_FILE_PATH=/certs/qms.chained.crt

qms-backend:
  # Configuration of SMTP server for sending alarm notification by email
  environment:
    # Hostname and port of the SMTP server
    SMTP_SERVER: ""
    SMTP_PORT: 1025
    # Credentials to use for connection to SMTP server
    SMTP_USERNAME: ""
    SMTP_PASSWORD: ""
    # Emails sent to distribution lists will have this sender
    SMTP_SENDER_NAME: "QMS"
    SMTP_SENDER_ADDRESS: "notif@example.com"

qms-postgres:
  volumes:
    - ./volumes/qms-postgres:/var/lib/postgresql/data:rw
```

MS Azure App Insights Integration



The QKD system also supports Microsoft Azure App Insights Integration for system statistics monitoring.

See user guide or CLI help index for additional information

MS Azure App Insights Integration

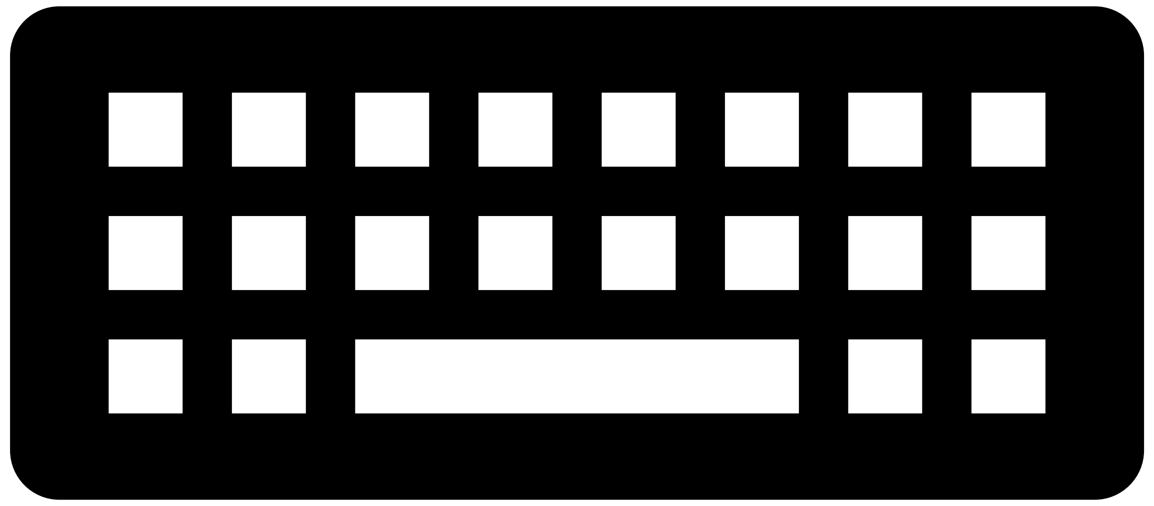
Azure AppInsights can also be enabled using QNET: See the following slides for additional QNET information.

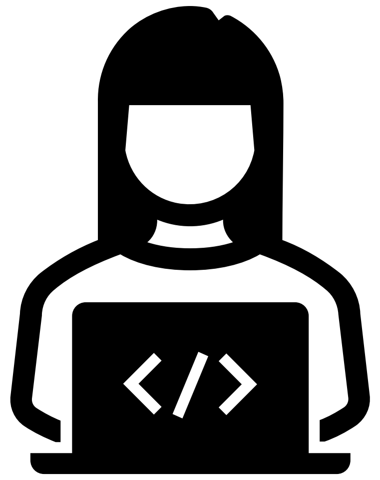
In order to forward logs & events from KMS or QKD managed by the QNET.WebAPI, use the following command, where:

- 12345678-1234-1234-1234-1234567890ab represents the Application Insights Instrumentation Key

qnet config qnetwebapi appinsights --enable --key 12345678-1234-1234-1234-1234567890ab
AppInsights settings successfully updated
QNET WebApi: AppInsights: Active
Instrumentation Key: 12345678-1234-1234-1234-1234567890ab

QNET Configuration API





QNet Configuration API

- QMS enables central management where administrators can configure and monitor systems and operation remotely.

- QMS provides:
  - a Web UI console (Discussed earlier)
  - a QNET (configuration) WebApi

- QNET administration tool is supported on Linux x64 platforms and Microsoft Windows 10 x64. For both platforms an installer is provided.

QNet Configuration API Cont'd

- QNET is a command line interface providing all commands to manage your KMS/QKD network configuration. You can then:
  - Build the configuration of your KMS/QKD network by using these commands through scripts
  - Export QMS configuration into XML/Json files for one node or a group of nodes
  - (Re) import XML/Json configuration into QMS
  - Deploy configuration file directly into KMS nodes without the need of QMS
  - Collect current configuration directly from KMS nodes

- QNET relies on 2 new components:
  - QNET WebApi embedded with QMS
  - QNC WebApi embedded in KMS

Sample QNet Commands (Script):

```
#!/bin/bash

# Authenticate First

qnet config qnetwebapi -url https://localhost:443 -user Admin -pwd Admin


#CREATE GROUP

echo "Creating new group.... (Deleting Group if it Already Exists)"

qnet delete group Group1

qnet create group Group1

echo "Creating node....."


#CREATE NODES

qnet create node CerberisXGA Group1 -type Mgmt -ip 10.10.10.102 -uid 8b03d09736e4482a8743479280c6ee20

qnet create node-network-interface CerberisXGA Key 192.168.10.1

qnet create node-network-interface CerberisXGA Kms 192.168.30.1


qnet create node CerberisXGB Group1 -type Mgmt -ip 10.10.10.107 -uid c9fa919a73cd456eb13969428e998861

qnet create node-network-interface CerberisXGB Key 192.168.10.2

qnet create node-network-interface CerberisXGB Kms 192.168.30.2


qnet create node-link --algorithm BlockCipher KMSLINK CerberisXGA CerberisXGB


#CREATE PROVIDERS

### PROVIDER MGMT IP REQUIRED FOR CERBERISXG-R

#qnet create provider QKDA CerberisXGA Cerberis3 -type Mgmt -ip 10.10.10.102

#qnet create provider-intf QKDA Any 192.168.20.1


### PROVIDER MGMT IP REQUIRED FOR CERBERISXG-R

#qnet create provider QKDB CerberisXGB Cerberis3 -type Mgmt -ip 10.10.10.107

#qnet create provider-intf QKDB Any 192.168.20.2


qnet create provider QKDA CerberisXGA Cerberis3 -type Any -ip 192.168.20.1

qnet create provider QKDB CerberisXGB Cerberis3 -type Any -ip 192.168.20.2


qnet create provider-link PROVLINK QKDA QKDB --key-req-interval 80


#CREATE CONSUMERS

qnet create consumer cisco CiscoA CerberisXGA --key-type QKEY --pport 9999

qnet create consumer cisco CiscoB CerberisXGB --key-type QKEY --pport 9999


qnet create consumer etsi014 ETSIA CerberisXGA --key-type QKEY --subject-dn "C=CH,ST=Geneva,L=Geneva,O=ID Quantique,OU=QuantumSafe,CN=QKDServer" --pport 443

qnet create consumer etsi014 ETSIB CerberisXGB --key-type QKEY --subject-dn "C=CH,ST=Geneva,L=Geneva,O=ID Quantique,OU=QuantumSafe,CN=QKDServer" --pport 443


qnet create consumer-link CiscoLINK CiscoA CiscoB --psk-ascii Cisco1234 --default-key-size 256

qnet create consumer-link ETSILINK ETSIA ETSIB --default-key-size 256


#CREATE PATHS

qnet create path CiscoA CiscoB -mode Automatic

qnet create path ETSIA ETSIB -mode Automatic


#ADD CERTIFICATES

qnet add ca-cert consumer --ca-file ChrisCA.pem

qnet add cert QKDServer.pem QKDServer_pkey.pem


qnet deploy group Group1


qnet update consumer etsi014 ETSIA CerberisXGA --cert-file QKDServer.pem --key-file QKDServer_pkey.pem

qnet update consumer etsi014 ETSIB CerberisXGB --cert-file QKDServer.pem --key-file QKDServer_pkey.pem


#DEPLOY THE FINAL CONFIG TO ALL NODES

qnet deploy group Group1
```

(Magnified on next slides)

Note that Clavis3 uses provider type "Cerberis3" and not "Clavis300".

Note: For V3.1.0, you must set the consumer-link default key size to 256

Sample QNet Commands (Script):

SWISS QUANTUM

```
#!/bin/bash
# Authenticate First
qnet config qnetwebapi -url https://localhost:443 -user Admin -pwd Admin

#CREATE GROUP
echo "Creating new group.... (Deleting Group if it Already Exists)"
qnet delete group Group1
qnet create group Group1
echo "Creating node....."

#CREATE NODES
qnet create node CerberisXGA Group1 -type Mgmt -ip 10.10.10.102 -uid 8b03d09736e4482a8743479280c6ee20
qnet create node-network-interface CerberisXGA Key 192.168.10.1
qnet create node-network-interface CerberisXGA Kms 192.168.30.1

qnet create node CerberisXGB Group1 -type Mgmt -ip 10.10.10.107 -uid c9fa919a73cd456eb13969428e998861
qnet create node-network-interface CerberisXGB Key 192.168.10.2
qnet create node-network-interface CerberisXGB Kms 192.168.30.2

qnet create node-link --algorithm BlockCipher KMSLINK CerberisXGA CerberisXGB
```

Sample QNet Commands (Script):

```
#CREATE PROVIDERS

qnet create provider QKDA CerberisXGA Cerberis3 -type Any -ip 192.168.20.1
qnet create provider QKDB CerberisXGB Cerberis3 -type Any -ip 192.168.20.2

qnet create provider-link PROVLINK QKDA QKDB --key-req-interval 80

#CREATE CONSUMERS
qnet create consumer cisco CiscoA CerberisXGA --key-type QKEY --pport 9999
qnet create consumer cisco CiscoB CerberisXGB --key-type QKEY --pport 9999

qnet create consumer etsi014 ETSIA CerberisXGA --key-type QKEY --subject-dn "C=CH,ST=Geneva,L=Geneva,O=ID Quantique,OU=QuantumSafe,CN=QKDServer" --pport 443
qnet create consumer etsi014 ETSIB CerberisXGB --key-type QKEY --subject-dn "C=CH,ST=Geneva,L=Geneva,O=ID Quantique,OU=QuantumSafe,CN=QKDServer" --pport 443

qnet create consumer-link CiscoLINK CiscoA CiscoB --psk-ascii Cisco1234 --default-key-size 256
qnet create consumer-link ETSILINK ETSIA ETSIB --default-key-size 256
```

Sample QNet Commands (Script):

```
#CREATE PATHS
qnet create path CiscoA CiscoB -mode Automatic
qnet create path ETSIA ETSIB -mode Automatic

#ADD CERTIFICATES
qnet add ca-cert consumer --ca-file ChrisCA.pem
qnet add cert QKDServer.pem QKDServer_pkey.pem

qnet deploy group Group1

qnet update consumer etsi014 ETSIA CerberisXGA --cert-file QKDServer.pem --key-file QKDServer_pkey.pem
qnet update consumer etsi014 ETSIB CerberisXGB --cert-file QKDServer.pem --key-file QKDServer_pkey.pem

#DEPLOY THE FINAL CONFIG TO ALL NODES
qnet deploy group Group1
```

Check if the system is working

Understand QKD logs

QKD logs

```
admin> help
1.1.32

Quantum Network Administrator
Administration utility to configure and manage your KMS/QKD components.


Usage: qnet [command] [options]

Options:
  --version   Show version information.
  --help      Show help information.

Commands:
  backup      backup current settings
  date        manage the datetime and timezone configuration
  exit        exit QNET shell
  help        displays help
  history     show the command history
  keylang     manage the keyboard language
  logcp       copy the system logs
  logtail     monitor the system logs
  network     manage the network configuration
  ntp         manage the NTP configuration
  password    change the current user's password
  qnc         manage the QNC configuration
  reboot      reboot the system
  reset       reset settings
  restart     restart a service
  restore     restore settings from USB
  shutdown    shutdown the system
  snmp        manage the SNMP configuration
  syslog      manage the SYSLOG configuration
  update      update the system
  version     show version information

Run 'qnet [command] --help' for more information about a command.

admin>
```

logtail **Monitor the system log**

→ Provide real-time view on the logs

logcp **Copy the system logs**

→ Copy logs on USB stick

QKD logs – logtail command

Logtail's command

"live" logs

```
admin> logtail qkd

2019-05-22T14:39:51.495 DEBUG [cervino::communicator::backend::OpticsRegulationAlice] QBER: 0.0161091
2019-05-22T14:39:51.496 DEBUG [cervino::communicator::backend::OpticsRegulationAlice] Visibility: 0.998145
2019-05-22T14:39:52.497 DEBUG [cervino::communicator::backend::OpticsRegulationAlice] QBER: 0.014082
2019-05-22T14:39:52.498 DEBUG [cervino::communicator::backend::OpticsRegulationAlice] Visibility: 0.995893
2019-05-22T14:39:53.216 INFO  [cervino::communicator::backend::KeyExchangeAlice] Key rate: 7500.8 bits/s
2019-05-22T14:39:53.490 DEBUG [cervino::communicator::backend::OpticsRegulationAlice] QBER: 0.012702
2019-05-22T14:39:53.490 DEBUG [cervino::communicator::backend::OpticsRegulationAlice] Visibility: 0.991526
2019-05-22T14:39:54.493 DEBUG [cervino::communicator::backend::OpticsRegulationAlice] QBER: 0.0201551
2019-05-22T14:39:54.494 DEBUG [cervino::communicator::backend::OpticsRegulationAlice] Visibility: 1.00309
2019-05-22T14:39:55.032 INFO  [cervino::communicator::backend::OpticsRegulationAlice] Regulating QBER: modulator bias = 5.31672 V, QBER = 0.0149385.
2019-05-22T14:39:55.053 INFO  [cervino::communicator::backend::OpticsRegulationAlice] Regulating visibility: laser current = 0.0472637 A, visibility = 0.996827.
2019-05-22T14:39:55.495 DEBUG [cervino::communicator::backend::OpticsRegulationAlice] QBER: 0.0172504
2019-05-22T14:39:55.496 DEBUG [cervino::communicator::backend::OpticsRegulationAlice] Visibility: 1.00114
2019-05-22T14:39:56.498 DEBUG [cervino::communicator::backend::OpticsRegulationAlice] QBER: 0.0164119
2019-05-22T14:39:56.498 DEBUG [cervino::communicator::backend::OpticsRegulationAlice] Visibility: 0.992503
2019-05-22T14:39:57.490 DEBUG [cervino::communicator::backend::OpticsRegulationAlice] QBER: 0.0192478
2019-05-22T14:39:57.491 DEBUG [cervino::communicator::backend::OpticsRegulationAlice] Visibility: 1.00274
2019-05-22T14:39:58.494 DEBUG [cervino::communicator::backend::OpticsRegulationAlice] QBER: 0.0156976
2019-05-22T14:39:58.494 DEBUG [cervino::communicator::backend::OpticsRegulationAlice] Visibility: 0.99852
2019-05-22T14:39:59.497 DEBUG [cervino::communicator::backend::OpticsRegulationAlice] QBER: 0.0147637
2019-05-22T14:39:59.498 DEBUG [cervino::communicator::backend::OpticsRegulationAlice] Visibility: 1.00538
2019-05-22T14:40:00.490 DEBUG [cervino::communicator::backend::OpticsRegulationAlice] QBER: 0.0138217
2019-05-22T14:40:00.491 DEBUG [cervino::communicator::backend::OpticsRegulationAlice] Visibility: 1.00936
2019-05-22T14:40:01.494 DEBUG [cervino::communicator::backend::OpticsRegulationAlice] QBER: 0.013315
2019-05-22T14:40:01.495 DEBUG [cervino::communicator::backend::OpticsRegulationAlice] Visibility: 1.00264
2019-05-22T14:40:02.496 DEBUG [cervino::communicator::backend::OpticsRegulationAlice] QBER: 0.0154504
2019-05-22T14:40:02.497 DEBUG [cervino::communicator::backend::OpticsRegulationAlice] Visibility: 1.00578
2019-05-22T14:40:03.219 INFO  [cervino::communicator::backend::KeyExchangeAlice] Key rate: 7321.6 bits/s
2019-05-22T14:40:03.499 DEBUG [cervino::communicator::backend::OpticsRegulationAlice] QBER: 0.0168633
2019-05-22T14:40:03.500 DEBUG [cervino::communicator::backend::OpticsRegulationAlice] Visibility: 1.00537
2019-05-22T14:40:04.491 DEBUG [cervino::communicator::backend::OpticsRegulationAlice] QBER: 0.0159271
2019-05-22T14:40:04.492 DEBUG [cervino::communicator::backend::OpticsRegulationAlice] Visibility: 0.994686
2019-05-22T14:40:05.347 INFO  [cervino::communicator::backend::OpticsRegulationAlice] Updating PA parameters: FullBlockHashValids: 512, FullBlockHashErrors:0
2019-05-22T14:40:05.347 INFO  [cervino::communicator::backend::OpticsRegulationAlice] Updating PA parameters: rawQBER = 0.020752, dccQBER = 0.0162114
2019-05-22T14:40:05.347 INFO  [cervino::communicator::backend::OpticsRegulationAlice] Updating PA parameters: rawVIS = 0.922685, dccVIS = 0.99514
.
```

QKD logs – Understand the logs

- Thousands of log lines are present, but you can easily find the information that you need.

- **Alice and Bob logs are different**, **they don't show the same type** of information

- You can use logtail <log> --pattern to filter by type of information

```
admin> logtail --help
monitor the system logs

Usage: logtail [options] <log>

Arguments:
  log                         log to monitor
                              Allowed values are: qkd, abrk, cagtc, cagtn, cagtr, clid, dbmd, ebrk, ecdhd, ibrk, kgnd, kmsd, pagti, qkrd, smmd, swd, symd,
                              qnet, webapi.

Options:
  --help                      Show help information.
  -p|--pattern <PATTERNS>     Monitor the system log with a specific pattern

admin>
```

- Logs lines car be either **INFO or DEBUG**

QKD logs – Understand the logs

- Before understanding the logs, we need to **understand the different steps** that are done during the system alignment phase

- If **one step is not properly finished** the system will **not go to** the **next step**

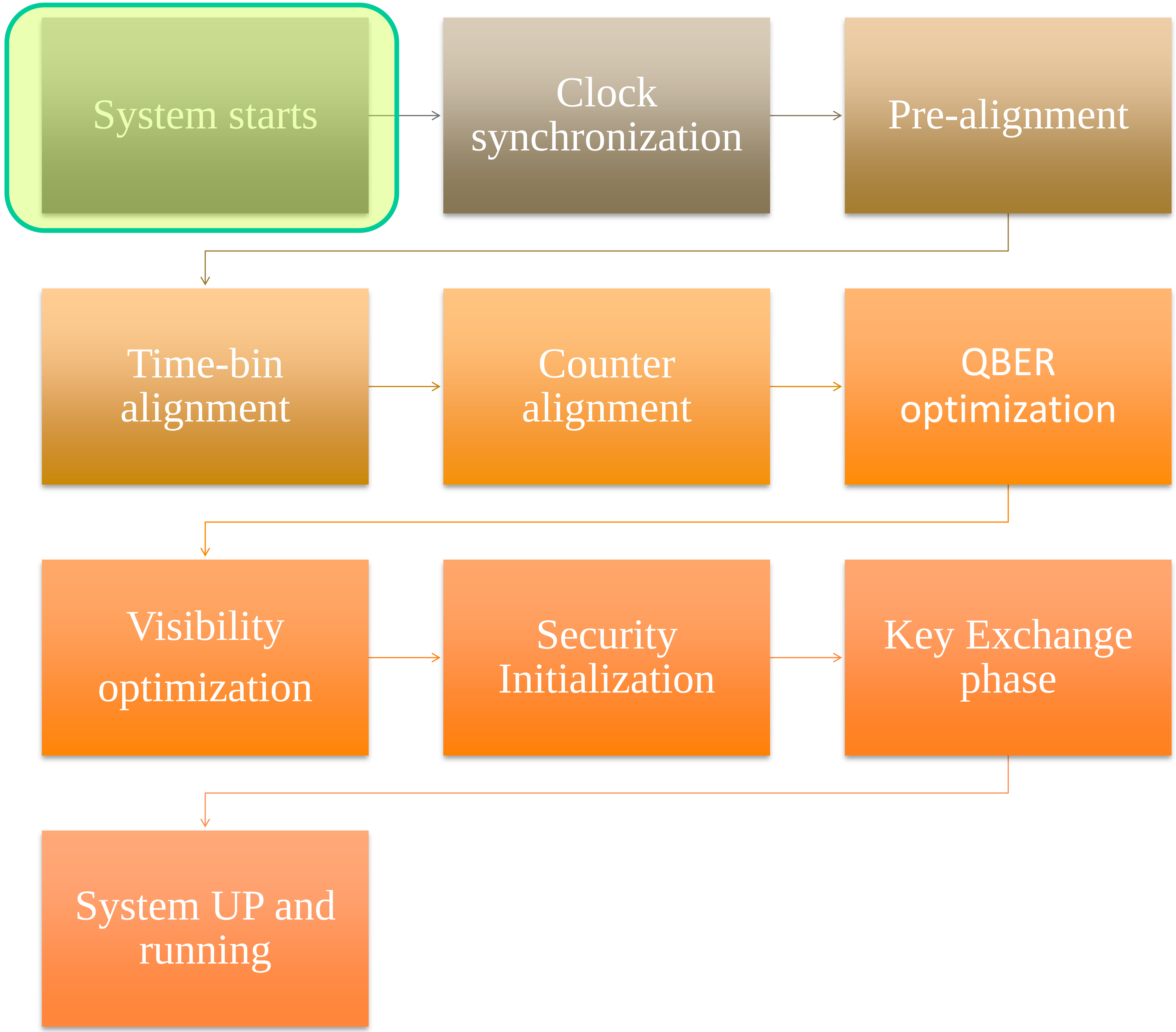
- Note there is a separate document titled “Understanding QKD Log Files” Please refer to this document for additional information.

QKD logs – Understand the QKD processes

SWISS QUANTUM

1. System starts →
2. Clock synchronization →
3. Pre-alignment →
4. Time-bin alignment →
5. Counter alignment →
6. QBER optimization →
7. Visibility optimization →
8. Security Initialization →
9. Key Exchange phase →
10. System UP and running

QKD logs – Understands the QKD processes



QKD logs – Understands the QKD processes

==	
Communicator Alice/Bob	
2.0.0.26678 RELEASE	Alice or Bob QKD Blade software version
Copyright (C) 2015-2021 ID Quantique SA	
==	
IDQ4P 1.4.0	IDQ4P version
ZeroMQ 4.3.4	ZeroMQ version
MsgPack 1.1.0	MessagePack version
==	

Subsequently, the log files show general system information:

System:	
Restart on failure: true	Whether to restat the system on failures
Max number of restart: 4	Maximum number of restarts before reboot
QKD timeout: 40 min	Time without QKD operation before failure state
IDQ4P:	
Interface: enp1s0f2	The network interface through used by the protocol
Encrypted: true	Whether the protocol is encrypted or not
Secret key: idq4p-communicator-default-skey.z85	Secret key used for the protocol encryption

QKD logs – Understands the QKD processes

SWISS QUANTUM

This is followed by general FPGA settings.

```
FPGA:
  Programmable trigger cycles:    125000000
  Parameter estimation:           true
  Distillation compression ratio: 30
  Authentication tag size:        8192
  Authentication fifo:            4095
  Model filter:                   true
  Protocol:                       COW
  QRNG enabled:                   true
  QRNG chip:                      true
  Deadtime mask:
0xf000000000000000
  DATA deadtime:                  5
  MONITOR deadtime:               5
  Base X probability:             10
  u0 probability:                 90
  u1 probability:                 10
  u2 probability:                 0
```

Programmable trigger cycles:	125000000	The FPGA frequency
Parameter estimation:	true	Determine the QBER computation mode
Distillation compression ratio:	30	Default compression ratio applied to the sifted and error-corrected key to obtain the secret key
Authentication tag size:	8192	
Authentication fifo:	4095	
Model filter:	true	Whether the keys are filtered according to the security validation by the internal model or not
Protocol:	COW	QKD protocol in use
QRNG enabled:	true	Whether the Quantum Random Number Generator is used to generated Qbit sequences or not
QRNG chip:	true	
Deadtime mask:	0xf000000000000000	(Bob only)
DATA deadtime:	5	(Bob only)
MONITOR deadtime:	5	(Bob only)
Base X probability:	10	(Alice only)
u0 probability:	90	(Alice only)
u1 probability:	10	(Alice only)
u2 probability:	0	(Alice only)

IDQ

QKD logs – Understands the QKD processes

Then, we find the general optical settings.

On Alice:

```
Optics:
  Pre-amplifier gain:                2.4
  Amplifier gain:                    1.16
  Laser current:                     0.05
  Laser target power photon number:  0.03
  Laser target power slope:          0.902649
  Laser target power intercept:      0.000314
  Pulse width:                       500
  Intensity modulator bias voltage:  [-11.5, 7.5]
```

On Bob:

```
Optics:
  DATA TEC target temperature:     -40
  DATA threshold:                  20
  DATA bias voltage:               66
  MONITOR TEC target temperature:  -40
  MONITOR threshold:               20
  MONITOR bias voltage:            66.8
  Detectors deadtime:              50
  External detectors:              false
  DATA pulse offset:               0
  DATA pulse width:                600
  MONITOR pulse offset:            0
  MONITOR pulse width:             600
```

QKD logs – Understands the QKD processes

The subsequent BIST section of the log file information summarizes the settings for the Built-In Self-Tests (BISTs).

On Alice:

```
BIST:
  Enabled:   true                         Whether BISTs are activated or not
  QrngBist: true
  LaserBist:
    Current:                              Laser current settings during the laser BIST phase
      offValue: 0
      offMin:    0
      offMax:    0.001
      onValue:   0.05
      onMin:     0.048
      onMax:     0.052
    PhotodiodePower:                      Expected power during the laser BIST phase
      offMin:    0
      offMax:    300
      onMin:     0.001
```

QKD logs – Understands the QKD processes

```
    onMax:     2300
  IMBist:
    Current:                         Laser current settings during the IM BIST phase
      onValue: 0.05
    IMBiasVoltages:                  IM bias voltage settings during the IM BIST phase
      Voltage: 0
      Voltage: -3
      Voltage: 3
    IMPhotodiodePowers:              Expected IM power during the IM BIST phase
      Power Min: 5 Max: 22
      Power Min: 5 Max: 22
      Power Min: 5 Max: 22
```

On Bob:

```
BIST:
  Enabled: true                                  Whether BISTs are activated or not
  QrngBist: true
  DetectorsBist:
    DATA TEC target temperature:    -40          Detectors temperature to reach during cooling
    MONITOR TEC target temperature: -40
    Timeout:                        900          Timeout for the detectors cool-down (in sec.)
```

QKD logs – Understands the QKD processes

SWISS QUANTUM

The subsequent section of the log files summarizes settings for the alignment phase..

On Alice (pre-alignment) :

```
Alignment:
  Photon number:          0.08
  IM bias voltage step: 1
  Period threshold:       1
```

Number of photons sent during alignment

IM bias voltage step size during alignment

On Bob:

```
Timebin Alignment:
  Trigger cycles:         31250000
```

Number of FPGA cycles per photon accumulation

```
  Min detections:         1000
  Detector 1 threshold: 0.9
  Detector 2 threshold: 0.9
Counter Alignment:
  Detector 1 max bit address: 21
  Detector 2 max bit address: 8
  Max counts:                 1024
  Threshold:                  0.4
```

Minimum number of photons to be detected in a frame

Correlation threshold for detector 1

Correlation threshold for detector 2

QKD logs – Understands the QKD processes

Only in Bob's log files, the subsequent information "Optics Optimization" concerns settings for the optics optimization phase:

```
QBER Minimization:
  Trigger cycles:        31250000              FPGA frequency
  Min detections:        1000                  Minimum number of detections to be accumulated
  DATA delay step:       1.1e-08               Detector 1 delay step used for the optimization
  IM bias voltage step:  0.05                  IM bias voltage step used for the optimization
Visibility Maximization:
  Trigger cycles:              31250000        FPGA frequency
  Min detections:              1000            Minimum number of detections to be accumulated
  MONITOR delay step:          1.1e-08         Detector 2 delay step
  Coarse laser current min:    -0.0015         Laser current min offset
  Coarse laser current max:    0.0015          Laser current max offset
  Coarse laser current step:   0.0001          Laser current step
  Fine laser current min:      -0.0001         Laser current min offset for fine adjustment
  Fine laser current max:      0.0001          Laser current max offset for fine adjustment
  Fine laser current step:     1e-05           Laser current step for fine adjustment
```

QKD logs – Understands the QKD processes

SWISS QUANTUM

On both Alice and Bob, we specify the timeout if the system generates no keys:

```
Key Exchange:

  Timeout: 20 min
```

Time with no key generation after which the system restarts

IDQ

QKD logs – Understands the QKD processes

The next information section "Optics Regulation" summarizes parameters related to the regulation of the optics during the key exchange, which is governed by Alice:

Optics Regulation:		
DATA dark counts:	694	Calibrated dark count rate of the DATA detector
DATA deadtime:	50	Common detector deadtime (in μs)
DATA detection efficiency:	0.2	Calibrated efficiency of the DATA detector
MONITOR dark counts:	691	Calibrated dark count rate of the MONITOR detector
MONITOR deadtime:	50	Common detector deadtime (in μs)
MONITOR detection efficiency:	0.2	Calibrated efficiency of the MONITOR detector
Bob's coupler ratio:	0.8	Splitting ratio of Bob's coupler
QBER integration time:	10	Integration time between QBER regulation steps (in sec.)
QBER epsilon:	0	N/A
Visibility integration time:	10	Integration time between visibility regulation steps (in sec.)
Visibility epsilon:	0	N/A
Modulator bias reduced step:	0.1	IM bias voltage step size of QBER regulation range
Modulator epsilon:	0.3	IM bias voltage jump size of QBER regulation range
Laser current min:	0.045	Min. laser current of visibility regulation range
Laser current max:	0.055	Max. laser current of visibility regulation range
Laser current period:	0.001	Laser current step size of visibility regulation range
Laser current jump multiplier:	5	Laser current jump size of visibility regulation range

On Bob's side we only specify the integration time for each regulation step:

Optics Regulation:	
Integration time: 10 sec	Integration time between regulation steps

QKD logs – Understands the QKD processes

After leaving the "PoweringOn" state, the system enters the Self-Test phase, which is indicated by the respective log file entry "Entering ExecutingSelfTest state".

Subsequently, the log files show the results of the Built-In Self Tests.

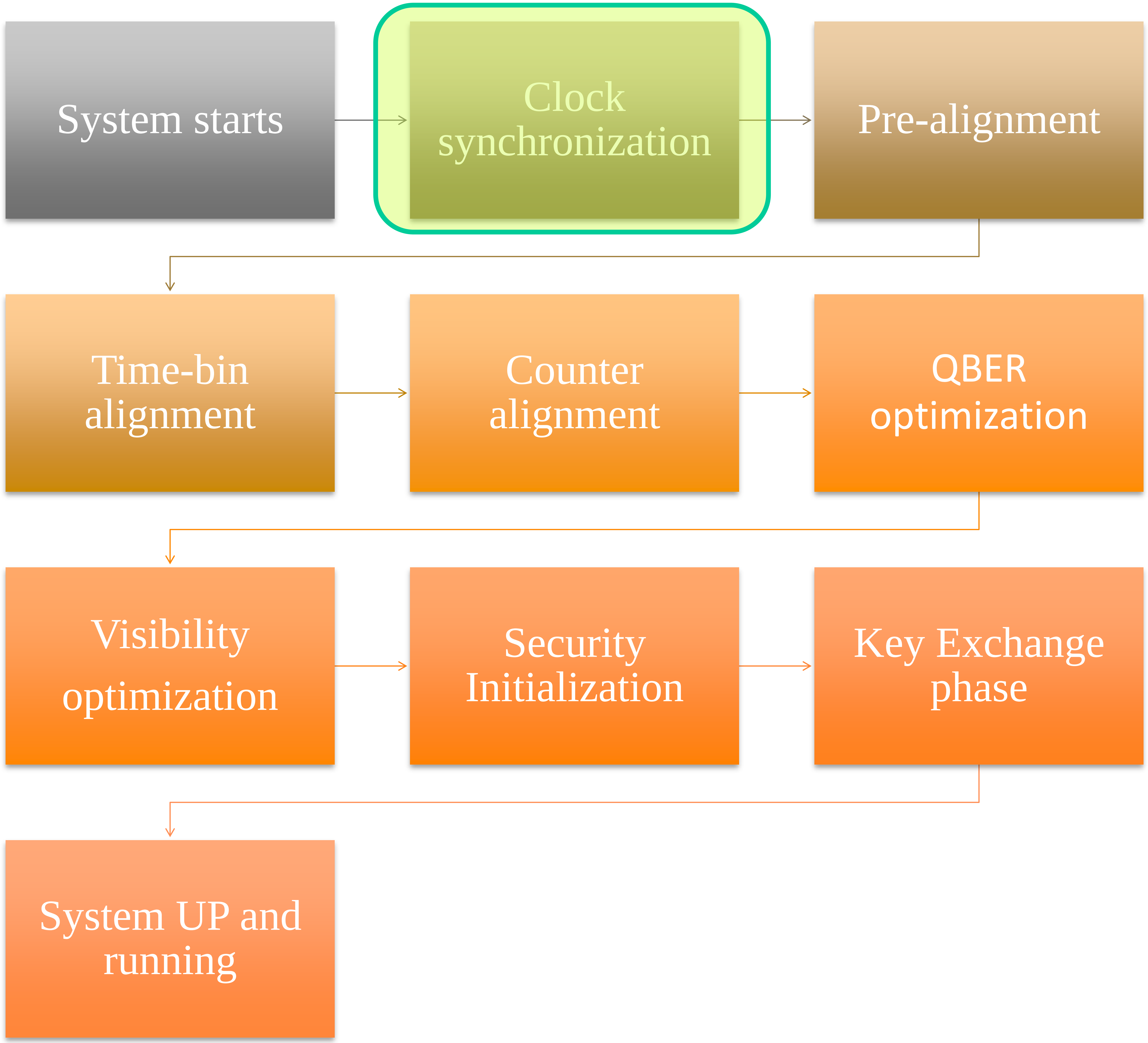
On Alice :

BIST Modules Version: PASSED	Check the version of the various modules
BIST BCP: PASSED	Check the communication with the Host board
BIST RCP-A: PASSED	Check the communication with the Alice board
BIST Laser Emission: PASSED	Check the quantum channel laser
BIST IM Extinction: PASSED	Check the quantum channel modulator
BIST QRNG: PASSED	Check the QRNG

On Bob:

BIST Modules Version: PASSED	Check the version of the various modules
BIST BCP: PASSED	Check the communication with the Host board
BIST RCP-B: PASSED	Check the communication with the Bob board
BIST Detectors Temperature: STARTED	
BIST Detectors Temperature: CHECK 1 PASSED	
BIST Detectors Temperature: CHECK 2 PASSED	Check the detectors' temperature
BIST Detectors Temperature: CHECK 3 PASSED	
BIST Detectors Temperature: PASSED	
BIST QRNG: PASSED	Check the QRNG

QKD logs – Understands the QKD processes



QKD logs – Understands the QKD processes

SWISS QUANTUM

Clock synchronization

At the beginning of this phase, Alice and Bob synchronize their clocks via the optical service channel, which is indicated in the log files by a sequence of entries in Alice's and Bob's log files:

`Local FPGA state:  SERVICE_CHANNEL_SYNC`	Current state of the local FPGA
`Remote FPGA state: SERVICE_CHANNEL_SYNC`	Current state of the remote FPGA
`Service channel synchronization done: false`	Indicates asynchronous FPGA clocks
`Local FPGA state:  SERVICE_CHANNEL_SYNC`	
`Remote FPGA state: SERVICE_CHANNEL_SYNC`	
`Waiting for clock recovery state...`	Indicates locking of the local PLL
`Service channel synchronization done: true`	FPGA clocks synchronized
`Local FPGA state:  IDLE`	

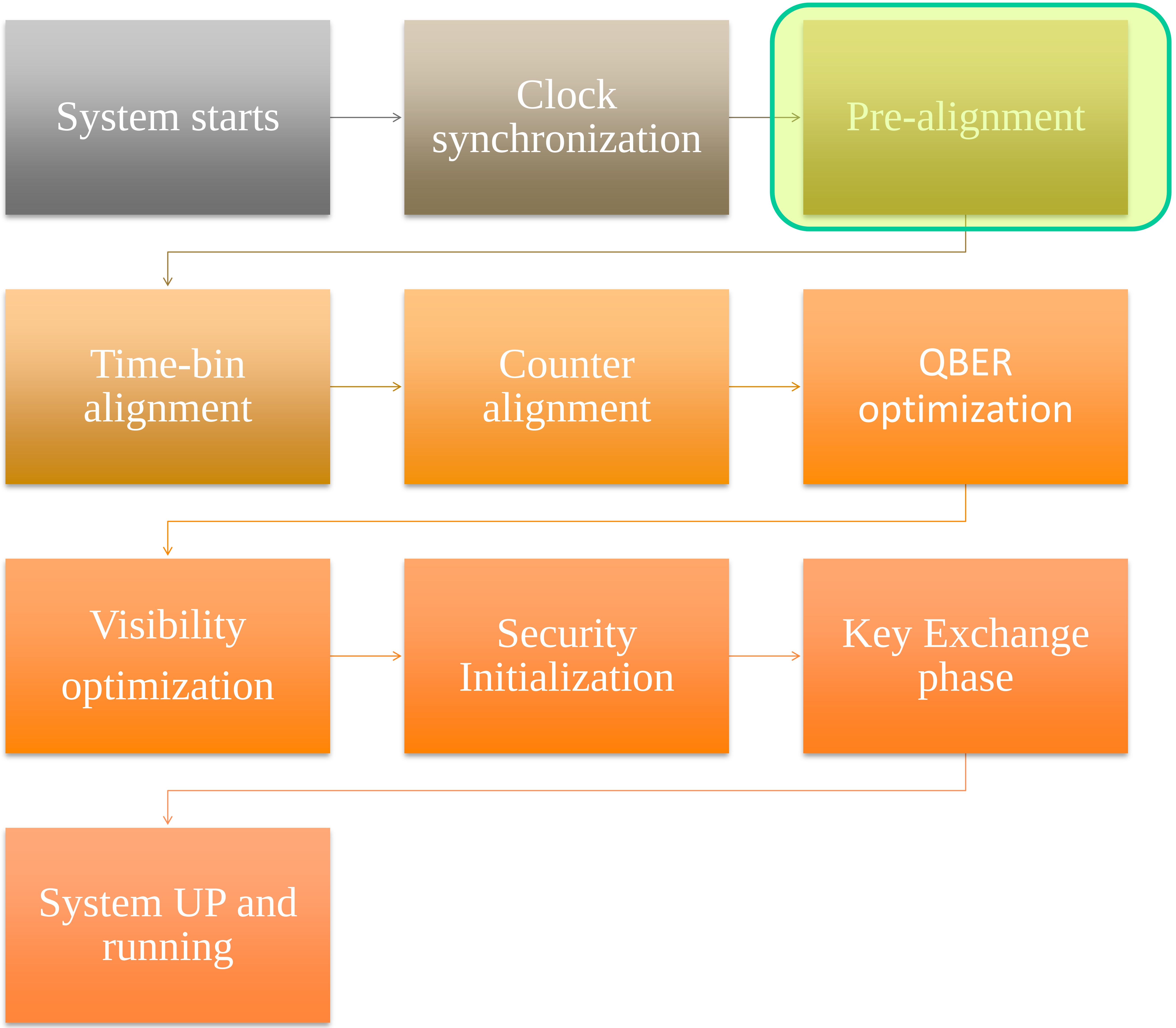
QKD logs – Understands the QKD processes

- If the FPGA clock is synchronized the system goes to the next step

```
Remote FPGA state: IDLE
Service channel status and states OK.
Service channel synchronized.
```

Synchronization of the FPGA clocks is achieved when the log files indicate for both, the local and the remote FPGA, to be in state "IDLE", and the clock recovery check passes.

QKD logs – Understands the QKD processes



QKD logs – Understands the QKD processes

- The IM voltage is scanned to find the best extinction-ratio

Pre-Alignment

After clock synchronization, the next initialization step is to pre-align Alice's modulator bias voltage. This is indicated in Alice's log file by a sequence of entries "Optimizing IM power" while scanning her IM bias voltage and measuring the variation in the optical power. If the measured optical power does not vary for different bias voltages, the modulator is not working properly. The pre-alignment is finished by Alice's log file entry "IM power optimized".

```
Pre-alignment pattern set to 0x88828
Optimizing IM power: bias voltage = -11.5 V, power = 0.0099687 A.
Optimizing IM power: bias voltage = -10.5 V, power = 0.0117867 A.
Optimizing IM power: bias voltage = -9.5 V, power = 0.014241 A.
Optimizing IM power: bias voltage = -8.5 V, power = 0.0165741 A.
Optimizing IM power: bias voltage = -7.5 V, power = 0.0183012 A.
Optimizing IM power: bias voltage = -6.5 V, power = 0.0192102 A.
Optimizing IM power: bias voltage = -5.5 V, power = 0.018786 A.
Optimizing IM power: bias voltage = -4.5 V, power = 0.0172104 A.
Optimizing IM power: bias voltage = -3.5 V, power = 0.014847 A.
Optimizing IM power: bias voltage = -2.5 V, power = 0.0123624 A.
Optimizing IM power: bias voltage = -1.5 V, power = 0.0102111 A.
Optimizing IM power: bias voltage = -0.5 V, power = 0.0089991 A.
Optimizing IM power: bias voltage = 0.5 V, power = 0.0091506 A.
Optimizing IM power: bias voltage = 1.5 V, power = 0.0105141 A.
Optimizing IM power: bias voltage = 2.5 V, power = 0.0129078 A.
Optimizing IM power: bias voltage = 3.5 V, power = 0.0155742 A.
Optimizing IM power: bias voltage = 4.5 V, power = 0.0179376 A.
Optimizing IM power: bias voltage = 5.5 V, power = 0.0193011 A.
```

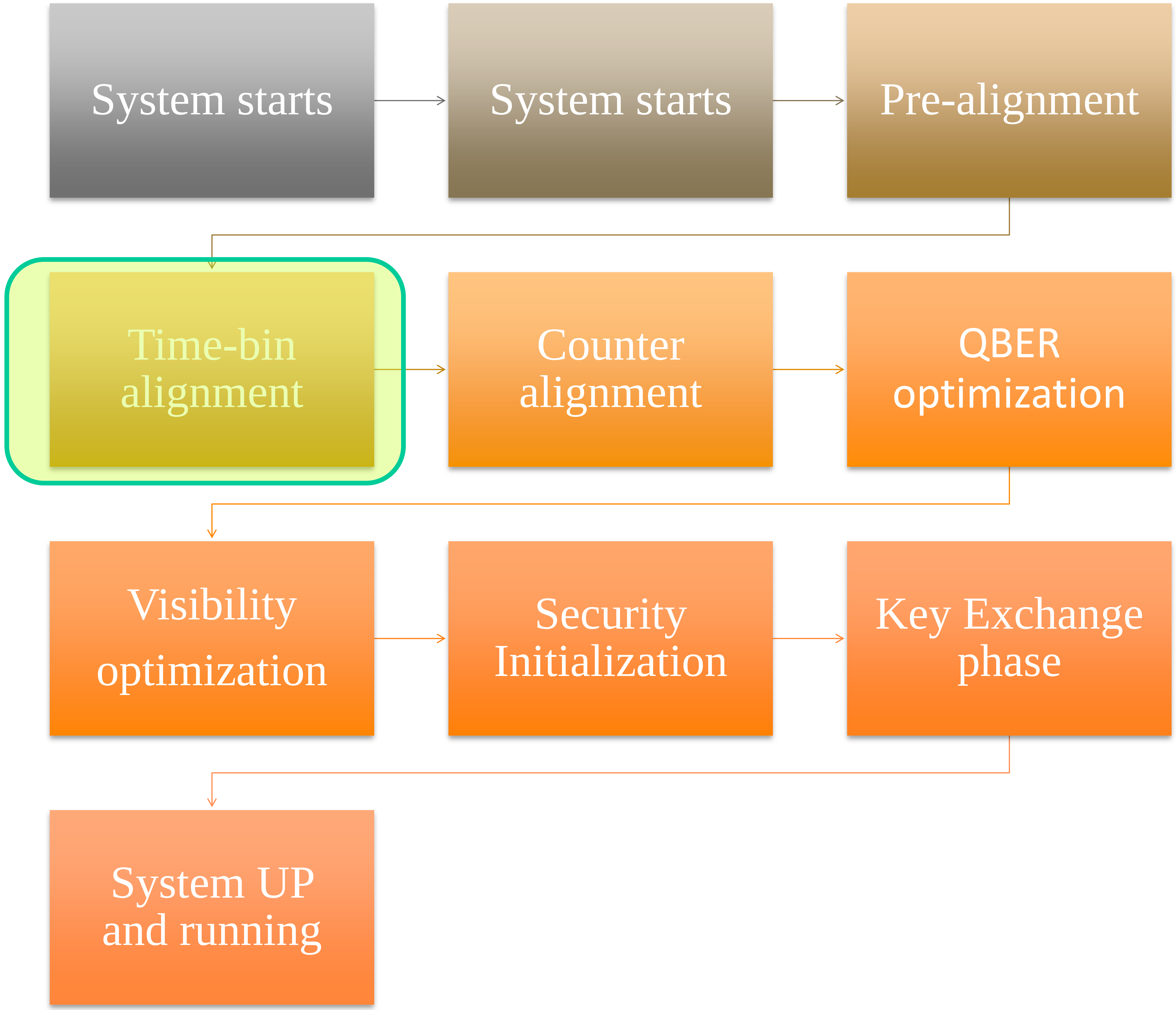
Coarse scan

QKD logs – Understands the QKD processes

```
Optimizing IM power: bias voltage = 6.5 V, power = 0.0194526 A.
Optimizing IM power: bias voltage = 7.5 V, power = 0.0182406 A.
Sinusoidal fit: offset = 0.0147788
Sinusoidal fit: amplitude = 0.00522675
Sinusoidal fit: frequency = 0.523599
Sinusoidal fit: phase = -1.5708
Sinusoidal fit: derivative error = 6.41575e-12
Sinusoidal fit: initial parameters = 0.00522675   0.523599    -1.5708  0.0147788
Sinusoidal fit: sum of squared residuals = 1.54326e-07
Sinusoidal fit: solution error = 0.0676655
Sinusoidal fit: inferred parameters = 0.00519041   0.502921   -1.50637  0.0141954
IM power: found highest IM power for bias voltage = 6.11858 V, power = 0.0193858 A.
```

Sinusoidal fit

QKD logs – Understands the QKD processes



QKD logs – Understands the QKD processes

SWISS QUANTUM

Time-bin alignment

After pre-alignment, the next initialization step is to find the beginning of a quantum channel frame by aligning the time-bins between Alice and Bob for Bob's DATA and MONITOR detector. While Bob performs the pattern matching, Alice's log files show the monitored optical power and the modulation pattern for time-bin alignment:

```
  Aligning DATA timebin...
1 Aligning DATA timebin: expecting pattern 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0
  | 0 | 0 | 0 | 0 | 0
  Aligning DATA timebin: non shifted timebins at delay = 2000:
2 66 | 18 | 5 | 19 | 620 | 893 | 65 | 19 | 5 | 18 | 558 | 919 | 79 | 18 | 3 | 12 | 20 | 9 | 598 |
  896
  …
3 Aligning DATA timebin: temporary best alignment for delay = 2000, shift = 1, correlation =
  0.81067.
  …
4 Aligning DATA timebin: found best alignment for delay = 2165, shift = 1, correlation = 0.994025.
  …
  Aligning MONITOR timebin...
1 Aligning MONITOR timebin: expecting pattern 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0
  | 1 | 0 | 0 | 0 | 0 | 0
  Aligning MONITOR timebin: non shifted timebins at delay = 2000:
2 397 | 85 | 19 | 19 | 18 | 27 | 457 | 99 | 474 | 88 | 17 | 33 | 504 | 96 | 461 | 85 | 24 | 29 | 488
  | 93
```

QKD logs – Understands the QKD processes

SWISS QUANTUM

```
…
3   Aligning MONITOR pulse: temporary best alignment for delay = 2000, shift = 14, correlation =
    0.984205.

…
4   Aligning MONITOR timebin: found best alignment for delay = 2330, shift = 13, correlation =
    0.997888.
```

After beginning with the time-bin alignment for the DATA detector, Bob's log files show the expected detection pattern (#1 in the figure above). Subsequently, Bob performs a scan of the detection delay, with his log file showing the measured pattern (#2) and the best alignment correlation found so far (#3).

NOTE

The maximum correlation factor must be above 0.9, otherwise the time-bin alignment will restart. If even after multiple trials no pattern with correlation factor above 0.9 is found, the QKD service automatically restarts. This process can be shortcut by manually restarting the QKD service using the CLI command restart.

QKD logs – Understands the QKD processes

NOTE

If the time-bin pattern (#2) is flat and doesn't show a clear detection pattern as in the figure above, then the quantum channel connection should be checked for excessive losses or bad connections. If this doesn't improve the pattern, the QKD service has to be restarted using the QKD-CLI command restart. *An exemplary flat pattern looks like this:*

```
141 | 216 | 190 | 325 | 262 | 267 | 170 | 158 | 161 | 172 | 193 | 183 | 201 | 178 | 138 | 222 | 234 | 221 | 259 | 178
```

QKD logs – Understands the QKD processes

SWISS QUANTUM

System starts → System starts → Clock synchronization

→ Time-bin alignment → Counter alignment → QBER optimization

→ Visibility optimization → Security Initialization → Key Exchange phase

→ System UP and running

IDQ

QKD logs – Understands the QKD processes

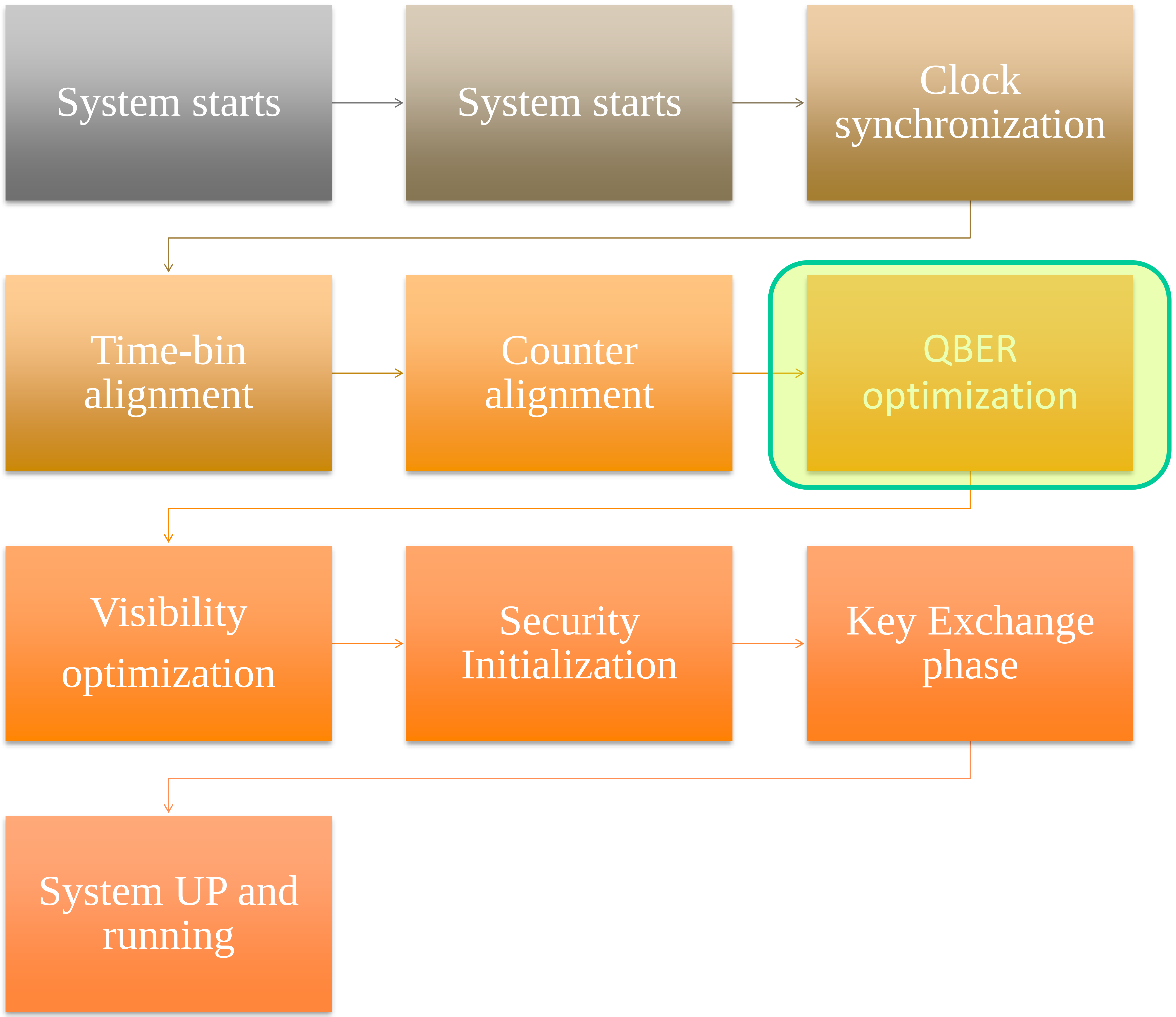
SWISS QUANTUM

```
Aligning DATA counter...
Aligning counter: alignment at bit 0
Aligning counter: XOR Counter 0 = 998, XOR Counter 1 = 26, correlation = 0.974609, QBER =
0.0253906
…
Aligning counter: alignment at bit 20
1 Aligning counter: XOR Counter 0 = 26, XOR Counter 1 = 998, correlation = 0.0253906, QBER =
0.974609
…
Aligning counter: bitflip bit 20
1 Aligning counter: XOR Counter 0 = 1002, XOR Counter 1 = 22, correlation = 0.978516, QBER =
0.0214844
DATA counter aligned.
Aligning MONITOR counter...
Aligning counter: alignment at bit 0
2 Aligning counter: XOR Counter 0 = 74, XOR Counter 1 = 950, correlation = 0.0722656, QBER =
0.927734
Aligning counter: bit 0 needs a bitflip.
Aligning counter: adding 1 Qbit delay.
Aligning counter: adding QbitDelay*4bit=4 bit delay.
3 Aligning counter: current clock delay = 0, shift delay = 13
Aligning counter: adding 4 bit delay
Aligning counter: new clock delay = 0, shift delay = 1
4 Aligning counter: XOR Counter 0 = 973, XOR Counter 1 = 51, correlation = 0.950195, QBER =
0.0498047
…
Aligning counter: alignment at bit 7
1 Aligning counter: XOR Counter 0 = 892, XOR Counter 1 = 132, correlation = 0.871094, QBER =
0.128906
MONITOR counter aligned.
```

Once the time-bins for the DATA and MONITOR detector have been aligned, the counter alignment for each detector follows. During this step, Alice's log file shows only the line "Aligning counter…", and after completion "Counter aligned.". At the same time, Bob's log files show the progress of the counter alignment:

IDQ

QKD logs – Understands the QKD processes



QKD logs – Understands the QKD processes

- QBER cannot exceed .045 maximum
- If QBER lower value is too high -> losses are too high, or problem with QC fiber connections

On Alice:

```
Minimizing QBER...
Minimizing QBER: modulator bias = 6.11261 V.
…
Minimizing QBER: modulator bias = 6.16261 V.
QBER minimized.
```

Currently applied modulator bias voltage


```
Minimizing QBER: found lowest QBER for modulator
bias = 6.16261 mV, DATA delay = 2156 ps, QBER =
0.0153846.
QBER minimized.
```

Min. QBER is found

On Bob:

```
Minimizing QBER...
Minimizing QBER: modulator bias = 6.11261 V.
Minimizing QBER: QBER = 0.0159744.
…
Minimizing QBER: QBER = 0.0162947.
Minimizing QBER: DATA delay = 2156 ps.
```

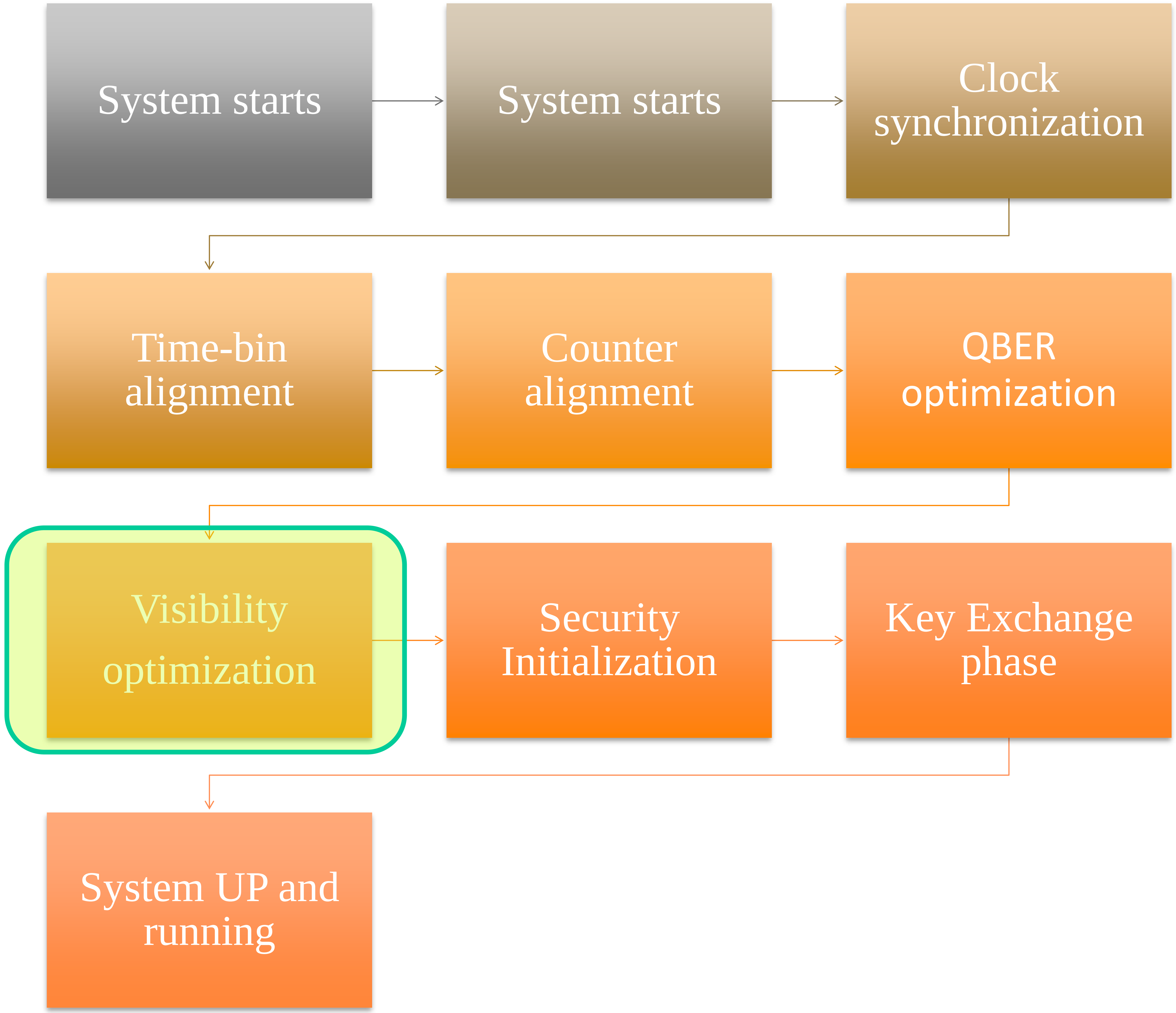
Currently applied modulator bias voltage and corresponding measured QBER

Currently applied DATA delay

```
Minimizing QBER: found lowest QBER for modulator
bias = 6.16261 mV, DATA delay = 2156 ps, QBER =
0.0153846.

QBER minimized.
```

Min. QBER is found

QKD logs – Understands the QKD processes



QKD logs – Understands the QKD processes

SWISS QUANTUM

- Visibility is scanned
- Final visibility must be close to 1.000 (i.e. > 0.95 is a good value)

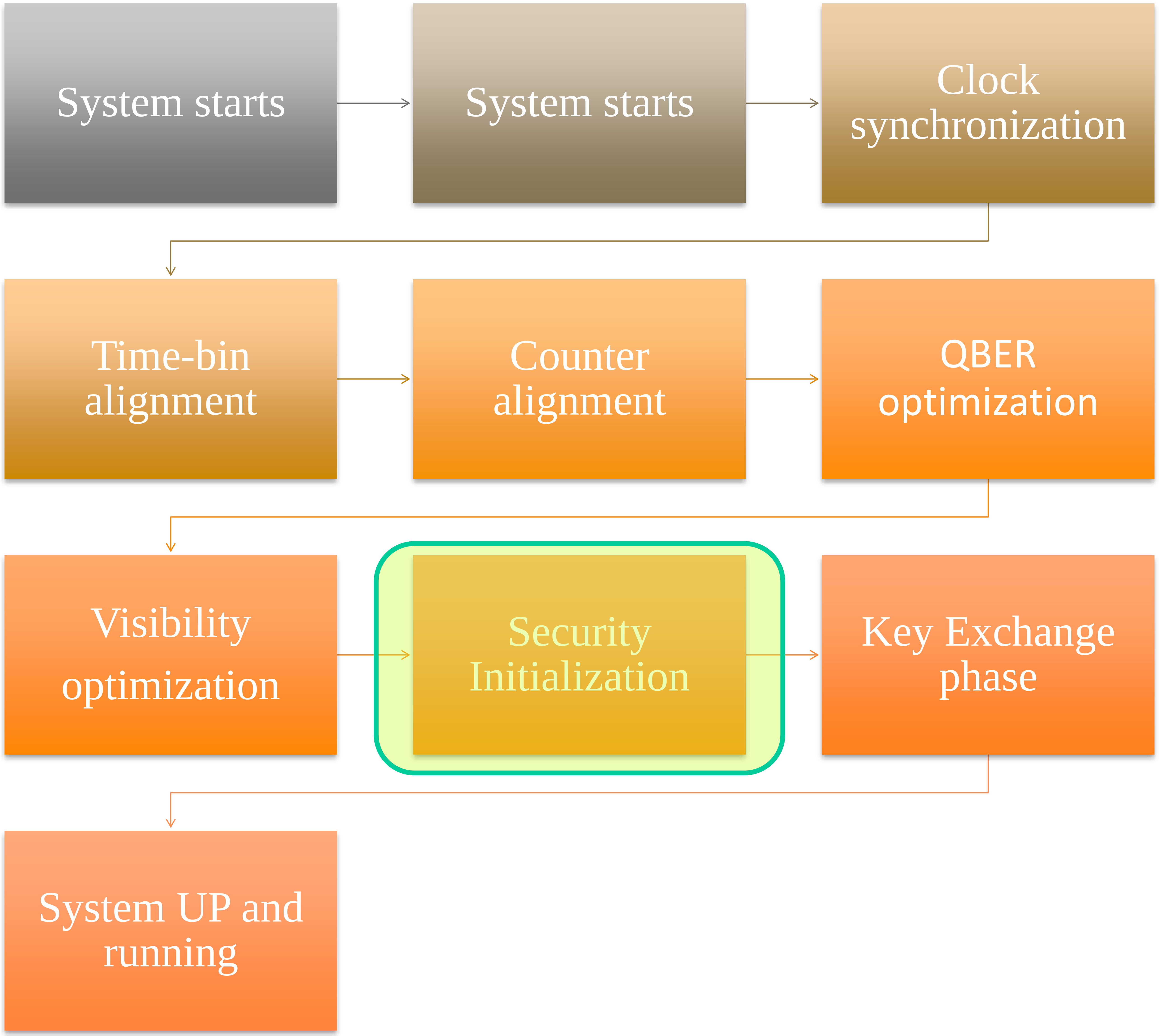
Alice

```
Maximizing visibility...                          Starting the Visibility Optimization step.
Maximizing visibility: current = 44999 uA.
Maximizing visibility: current = 43499 uA.        Currently applied laser bias current.
⋮
Maximizing visibility: current = 44639 uA.
Visibility maximized.                             Finalizing the Visibility Optimization.
Leaving ExecutingGeneralInitialization state.     Finalizing the General Initialization phase.
```

Bob

```
Maximizing visibility...                                     Starting the Visibility Optimization step.
Remote FPGA Optics Optimization state: 1 (expected 2).       Waiting for Alice.
Maximizing visibility: changing laser current value...
Maximizing visibility: current = 44999 uA.
Maximizing visibility: current = 43499 uA.
Maximizing visibility: visibility = 0.467572.                Currently applied laser bias current and
⋮                                                            corresponding measured Visibility.
Maximizing visibility: current = 45999 uA.
Maximizing visibility: visibility = 0.966131.
Maximizing visibility: found highest visibility for …
Visibility maximized.                                        Finalizing the Visibility Optimization.
Leaving ExecutingGeneralInitialization state.                Finalizing the General Initialization phase.
```

QKD logs – Understands the QKD processes



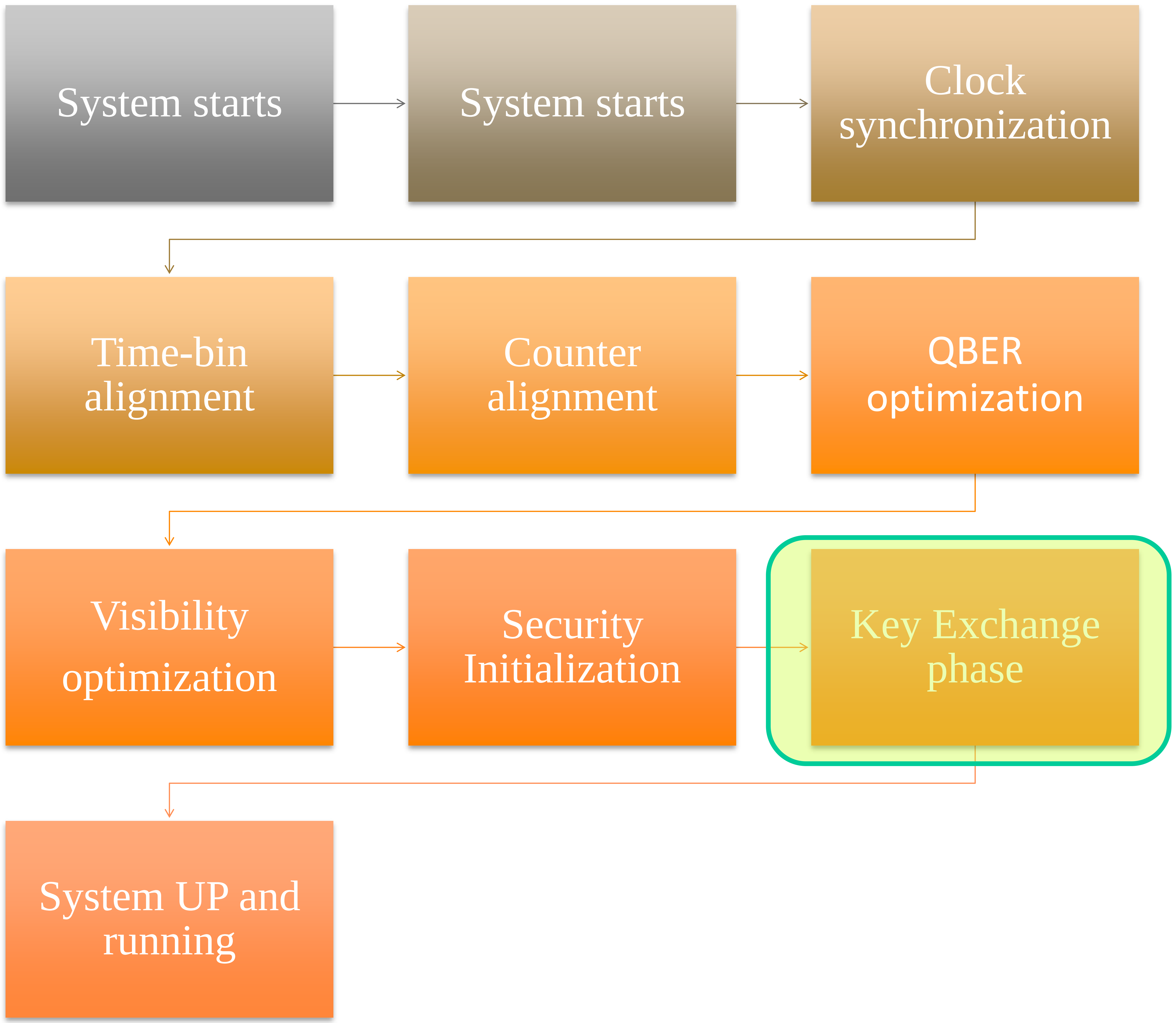
QKD logs – Understands the QKD processes

- If the initial key is not provided, you should verify the QMS configuration has been deployed.

- If needed you could re-create and re-deploy the path.

Entering ExecutingSecurityInitialization state.	Starting the Security Initialization phase.
Received IDQ4P command 'SetInitialKey'.	Receiving the Initial Key from the QNC.
Processing event SetInitialKey.	Processing the Initial Key.
Setting initial authentication key...	Setting the Initial Key.
Initial authentication key set.	Initial Key set.

QKD logs – Understands the QKD processes



QKD logs – Understands the QKD processes

- If the system cannot generate keys after aligning 4x in a row, **the system reboots**
- The key rate could stay at 0 bits/sec for few minutes, until enough qubits are accumulated for the key distillation process.

Alice

Entering Running state.	Starting the Key Exchange.
Key exchange started.	Key exchange started.
QBER: 0.021227	Currently measured QBER.
Visibility: 0.966967	Currently measured Visibility.
...	
Regulating QBER: modulator bias = 6.63949 V, QBER = 0.0119892.	QBER regulation.
Regulating visibility: laser current = 0.044639 A, visibility = 0.958673.	Visibility regulation.
...	
Updating PA parameters: FullBlockHashValids: 512, FullBlockHashErrors:1	Hash errors.
Updating PA parameters: rawQBER = 0.0229984, dccQBER = 0.0134074	Raw and dark count corrected QBER.
Updating PA parameters: rawVIS = 0.881038, dccVIS = 0.958	Raw and dark count corrected Visibility.
Updating PA parameters: mu = 0.0300399, cr = 0.0956185	Photon number and new compression ratio.
Updating PA parameters: Timer = 51.52	Key accumulation time (in seconds).
Updating PA parameters: Column: 4, Row: 2	
Next compression ratio: 0.0956185	New compression ratio for next key.
Current compression ratio: 0.0952289	Currently applied compression ratio.
...	

Bob

Entering Running state.	Starting the Key Exchange.
Starting key exchange...	Key exchange started.
Key rate: 0 bits/s	Current secret key rate.
Regulating detection ratio: ratio = 1.25979, DATA delay = 2189 ps, MONITOR delay = 2365 ps.	Detector delay regulation.
...	
Key rate: 0 bits/s	
Regulating detection ratio: ratio = 1.11457, DATA delay = 2178 ps, MONITOR delay = 2354 ps.	Detector delay regulation.
Next compression ratio: 0.0956185	New compression ratio for next key.
Current compression ratio: 0.0952289	Currently applied compression ratio.
Key rate: 2227.2 bits/s	Current secret key rate.
...	

CerberisXG/Cerberis3

SNMP monitoring

Management and monitoring

- CerberisXG/QKD and QNC parameters and realtime values can be checked using:

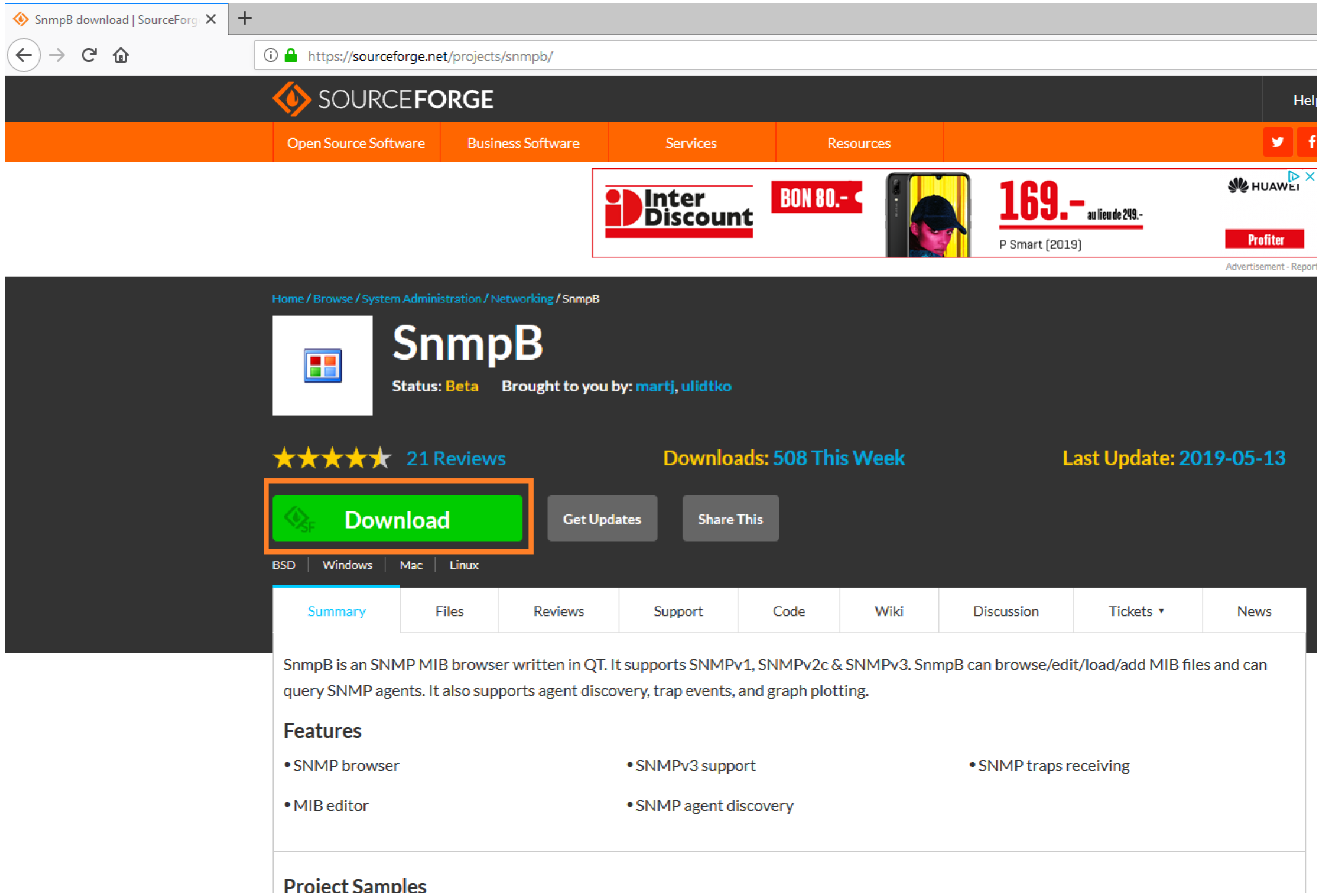
    - SNMPv3 walk
    - Syslogs
    - Logs
    - QMS

- The recommended lab tool to use snmp walk is SNMPB (open source)

SNMPB configuration

1. Download and Install SNMPB

- Go to https://sourceforge.net/projects/snmpb/
- Download the file
- Install the tool

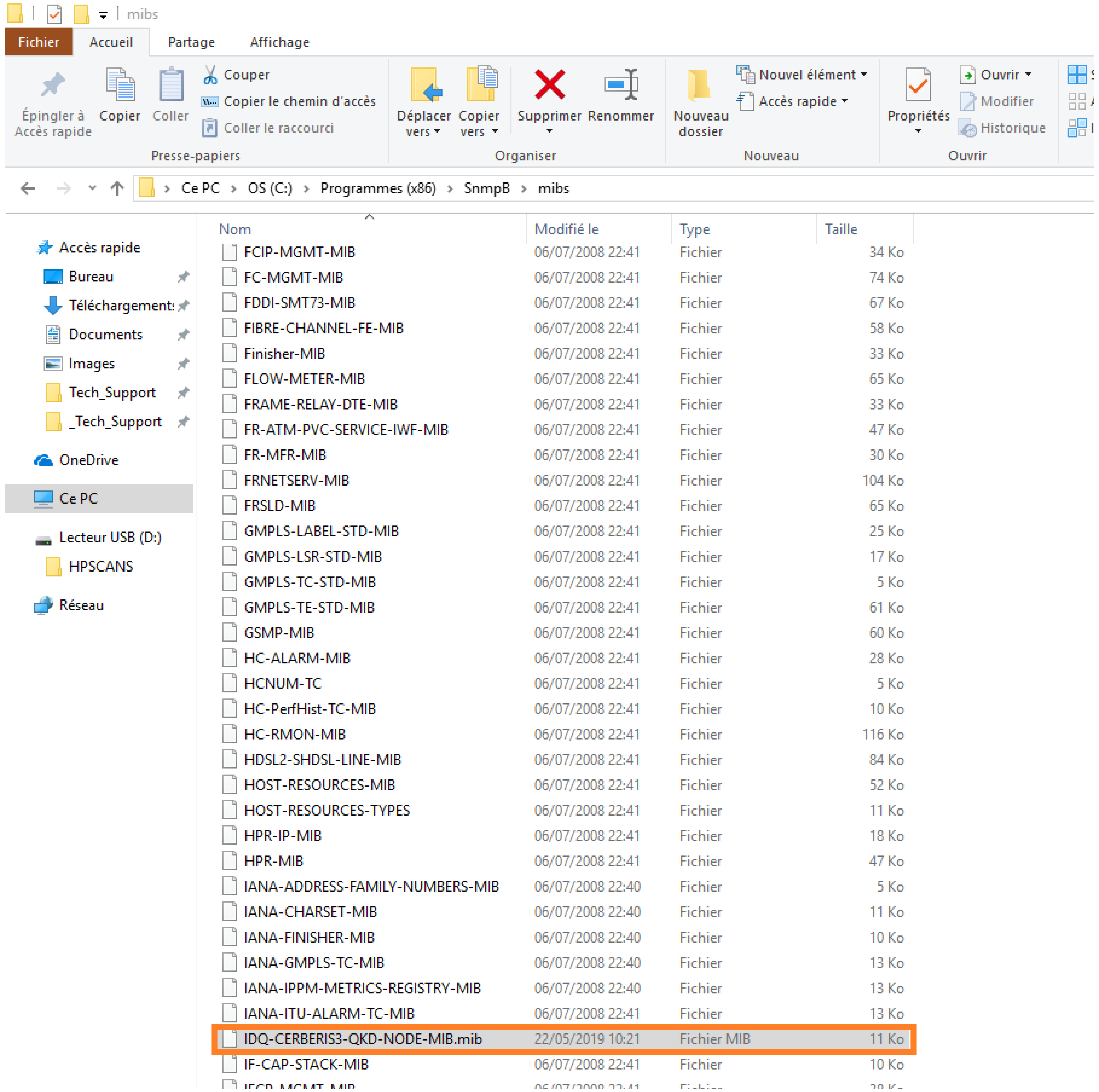


SNMPB configuration

SWISS QUANTUM

2. Copy MIB file to root directory

- Get the MIB files named:
  1. IDQ-CERBERIS3-QKD-NODE-MIB.mib
  2. VT-SH.mib
- Copy files to to C:\Program Files (x86)\SnmpB\mibs

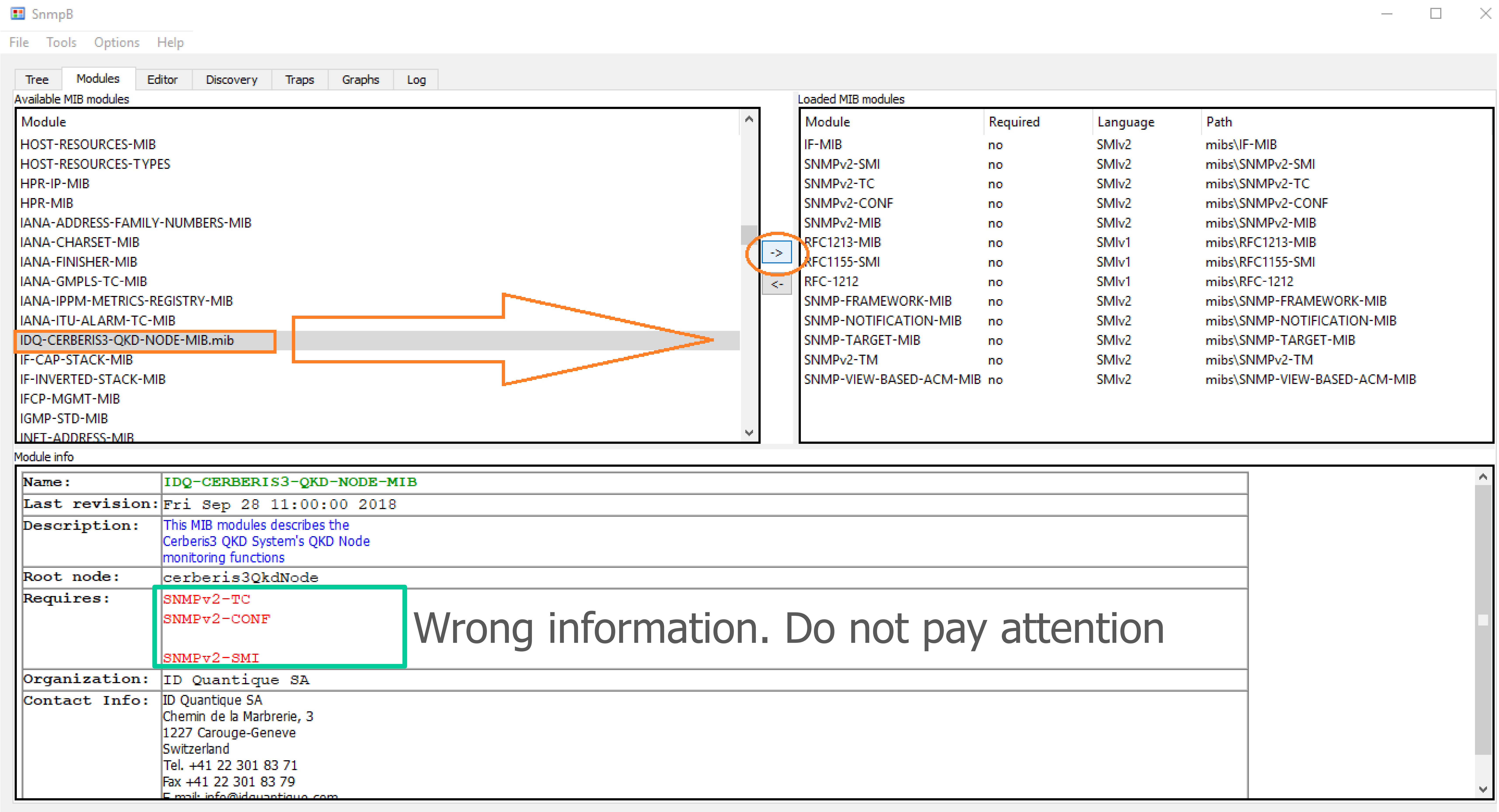


IDQ

SWISS QUANTUM

3. Configure MIBs on SNMPB

- Open SNMPB
- Move the MIB files from “Available MIB Modules” to “Loaded MIB modules”



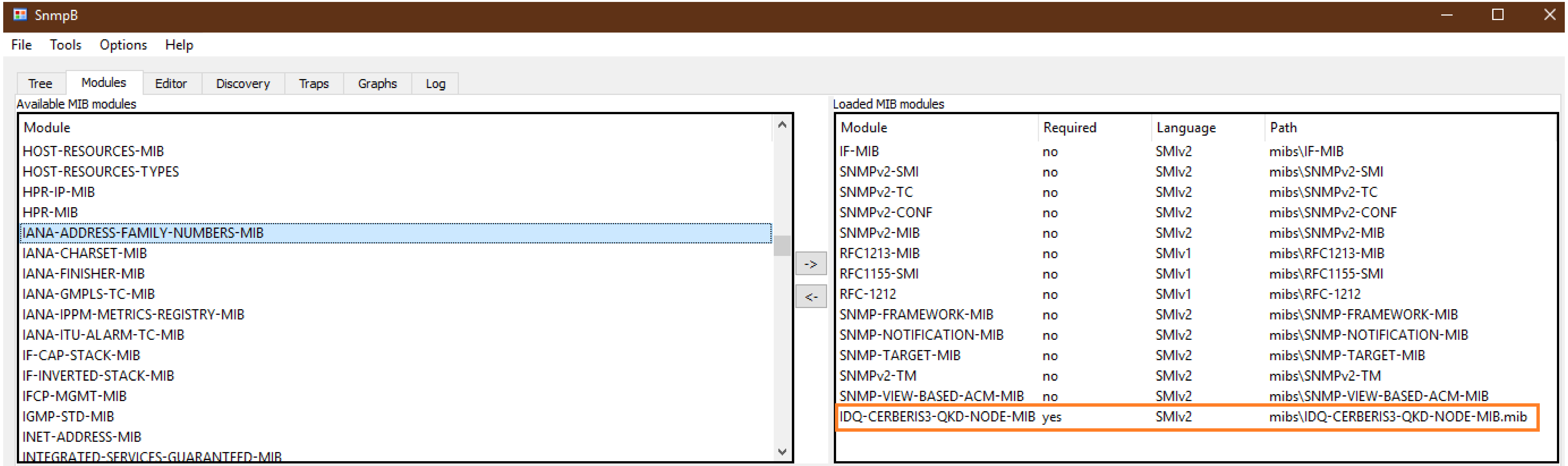
Move Both MIB Files!

IDQ

SNMPB configuration

3. Configure MIBs on SNMPB

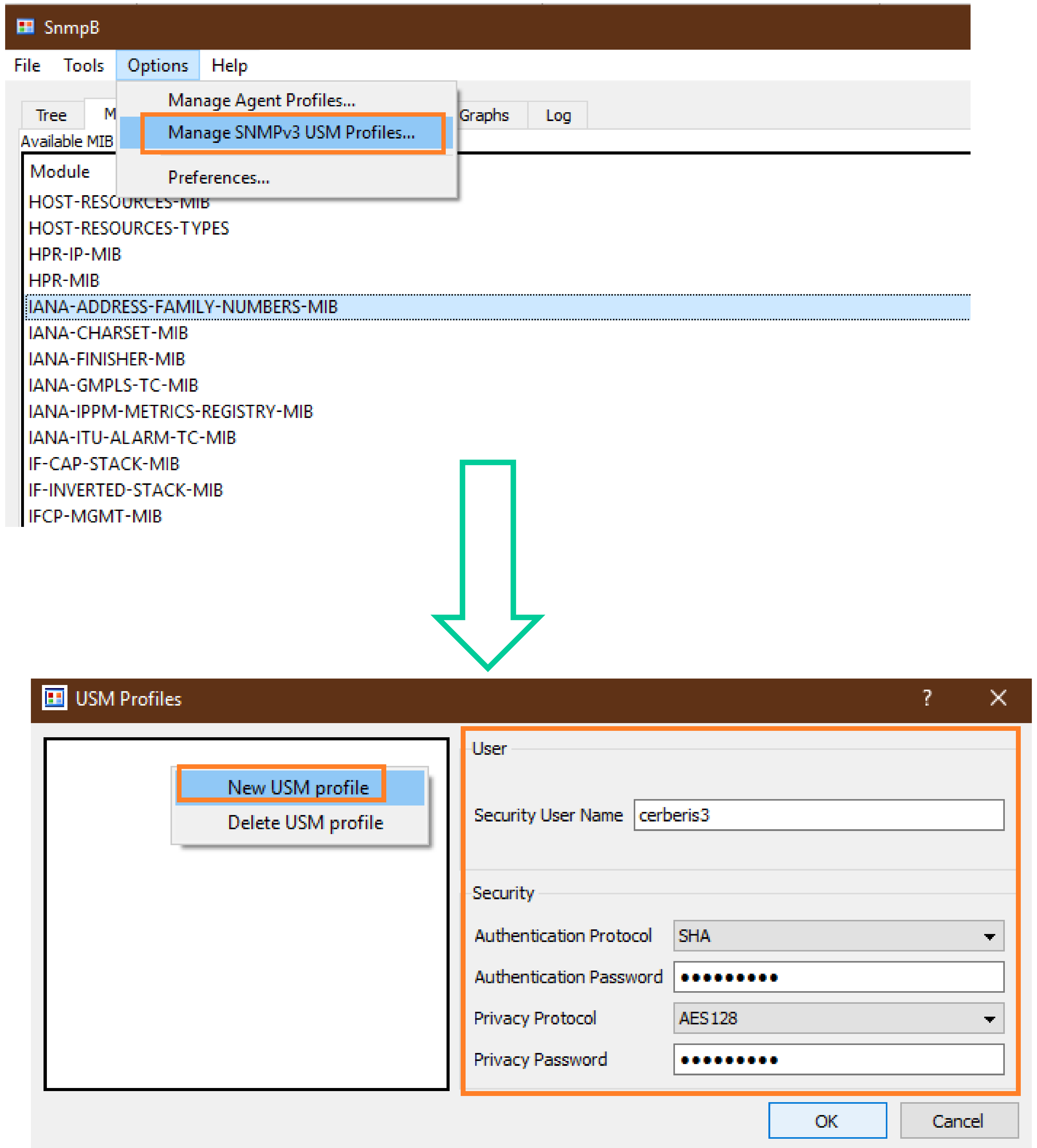
- The MIBs should appear to the right side (Loaded MIB modules)



SNMPB configuration

4. Create SNMPv3 user

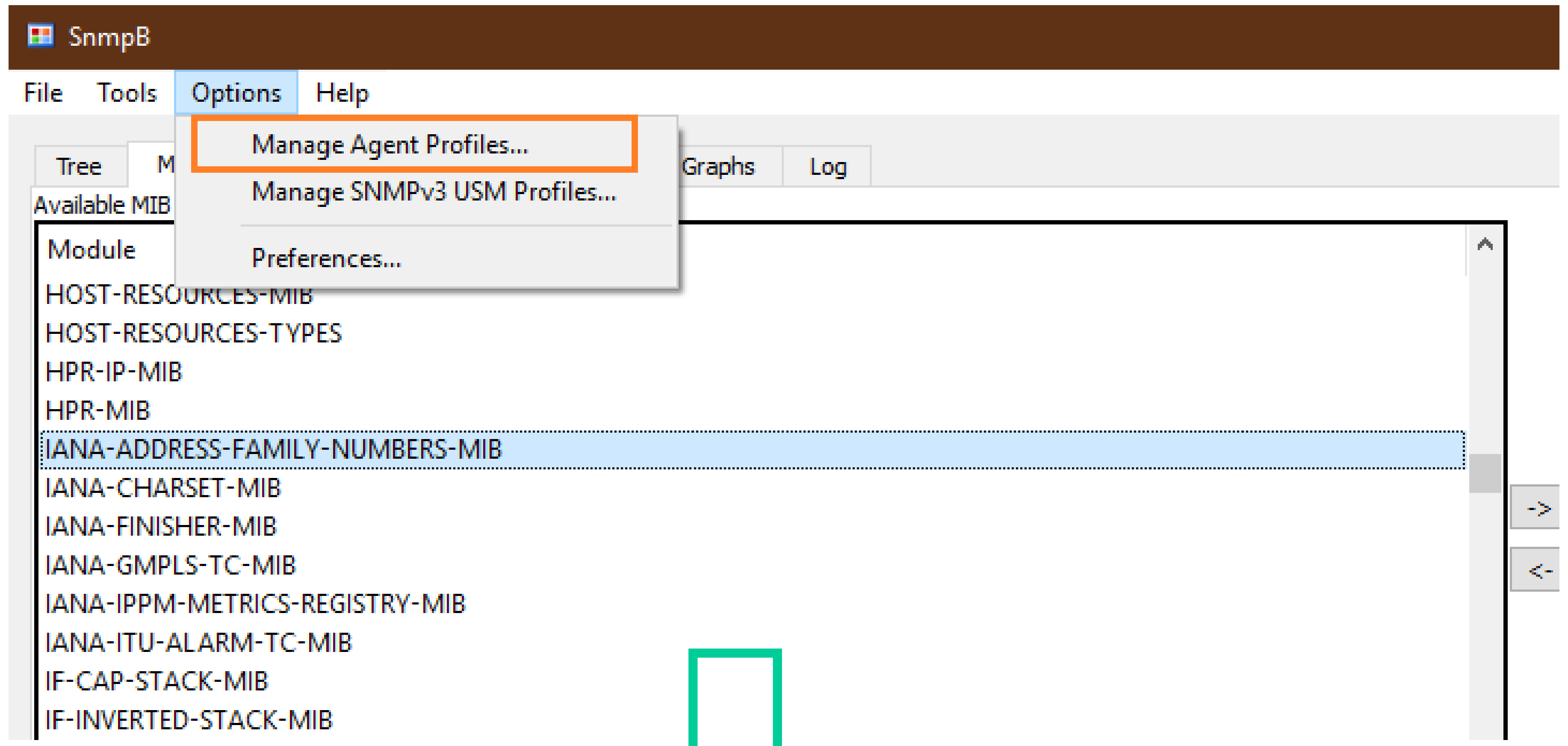
- Go to Option / Manage SNMPv3 USM Profiles
- Do a right click on the blank zone and select “New USM Profile”
- Insert the right values:
  - Set Security User Name to cerberis3.
  - Set Authentication Protocol to SHA.
  - Set Authentication Password to cerberis3.
  - Set Privacy Protocol to AES128.
  - Set the Privacy Password to cerberis3.

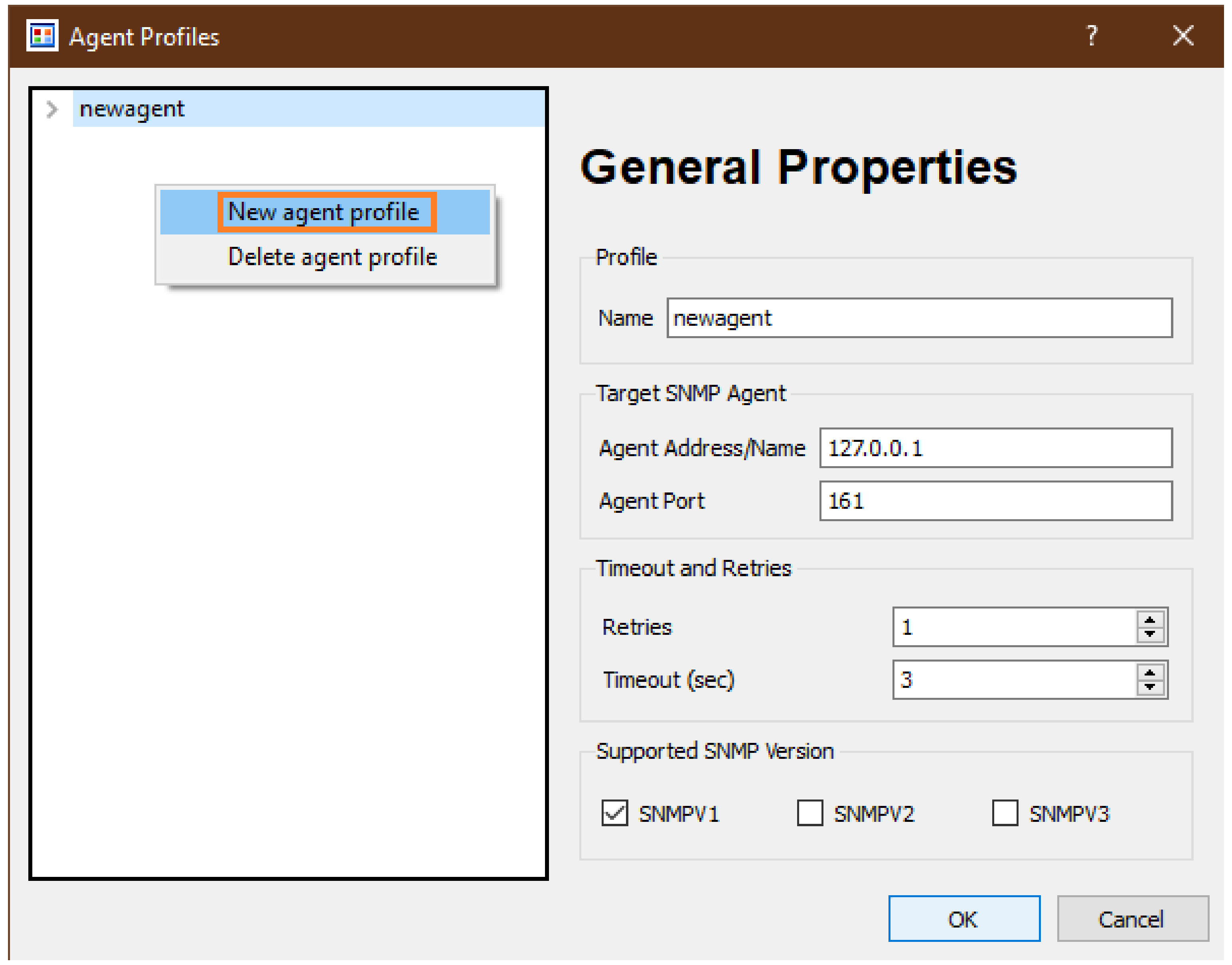


SNMPB configuration

5. Create Agent Profile

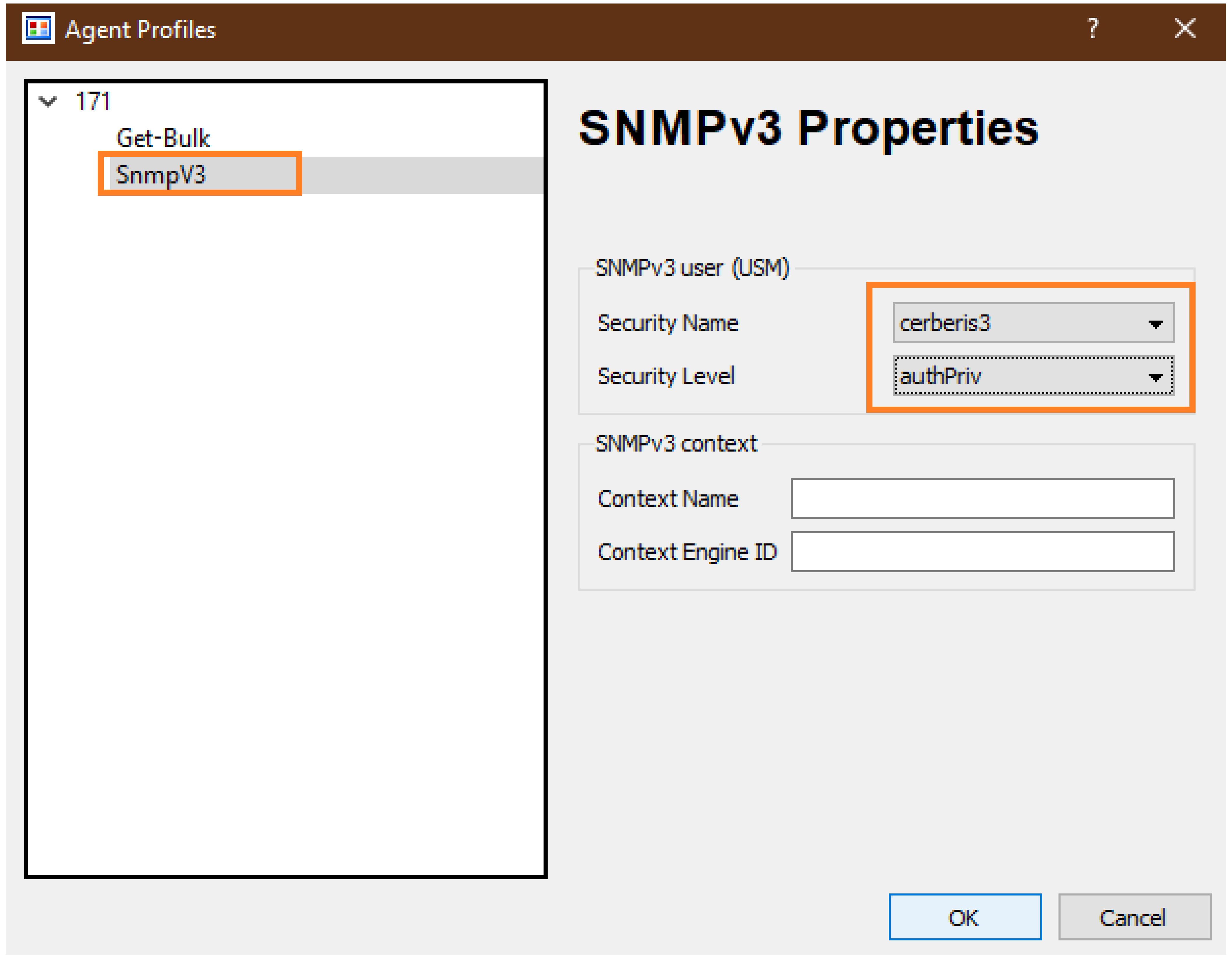
- Go to Option / Manage Agent Profiles
- Do a right click on the blank zone and select “New agent profile”
- Insert the right values:
    - Name: As you want.
    - Agent Address/Name: QNC IP Address
    - Agent port: 161
    - Retries: 1
    - Timeout: 3
    - Supported SNMP Version: SNMPv3





6. Configure SNMPv3 on Agent Profile

- Do a right click on the Profile Name
- Select SNMPv3
- Select “cerberis3” for Security Name
- Select “authPriv” for Security Level



Configure SNMP Parameters on each CerberisXG/QNC

SWISS QUANTUM

```
admin> snmp --user cerberis3
The new SNMP configuration was successfully applied.
admin> snmp --authPassword cerberis3
The new SNMP configuration was successfully applied.
admin> snmp --privPassword cerberis3
The new SNMP configuration was successfully applied.
admin>
```

Configure 2x:
Once for each
CerberisXG/QNC

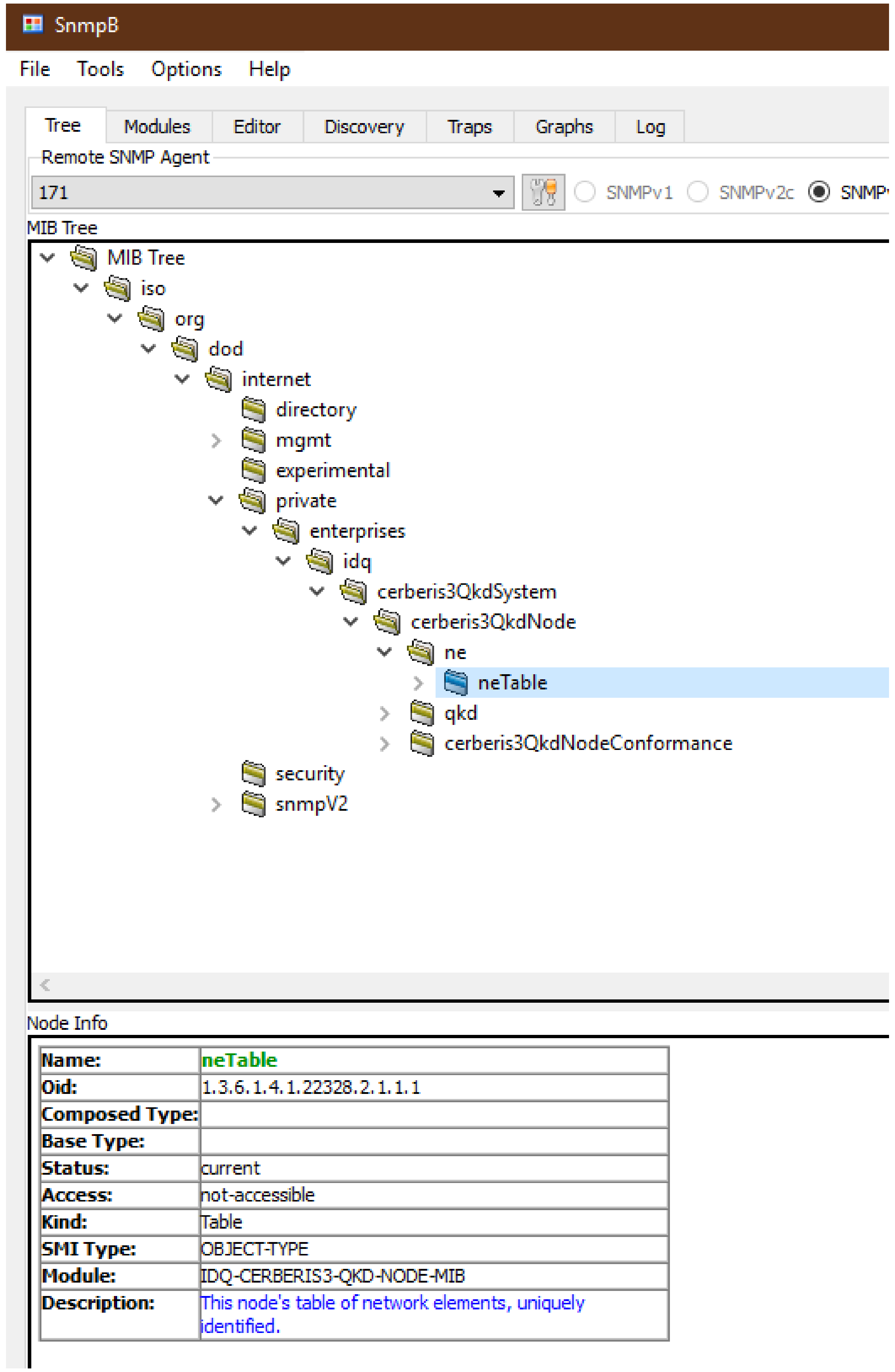
Help

```
admin> snmp --help
manage the SNMP configuration

Usage: snmp [options]

Options:
  --help                    Show help information.
  -c|--contact <arg>        Set the system contact
  -n|--name <arg>           Set the system name
  -l|--location <arg>       Set the system location
  -u|--user <arg>           Set the user name
  -a|--authPassword <arg>   Set the user authentication password
  -x|--privPassword <arg>   Set the user encryption password

admin>
```

SNMPB configuration

7. Check if the communication is working

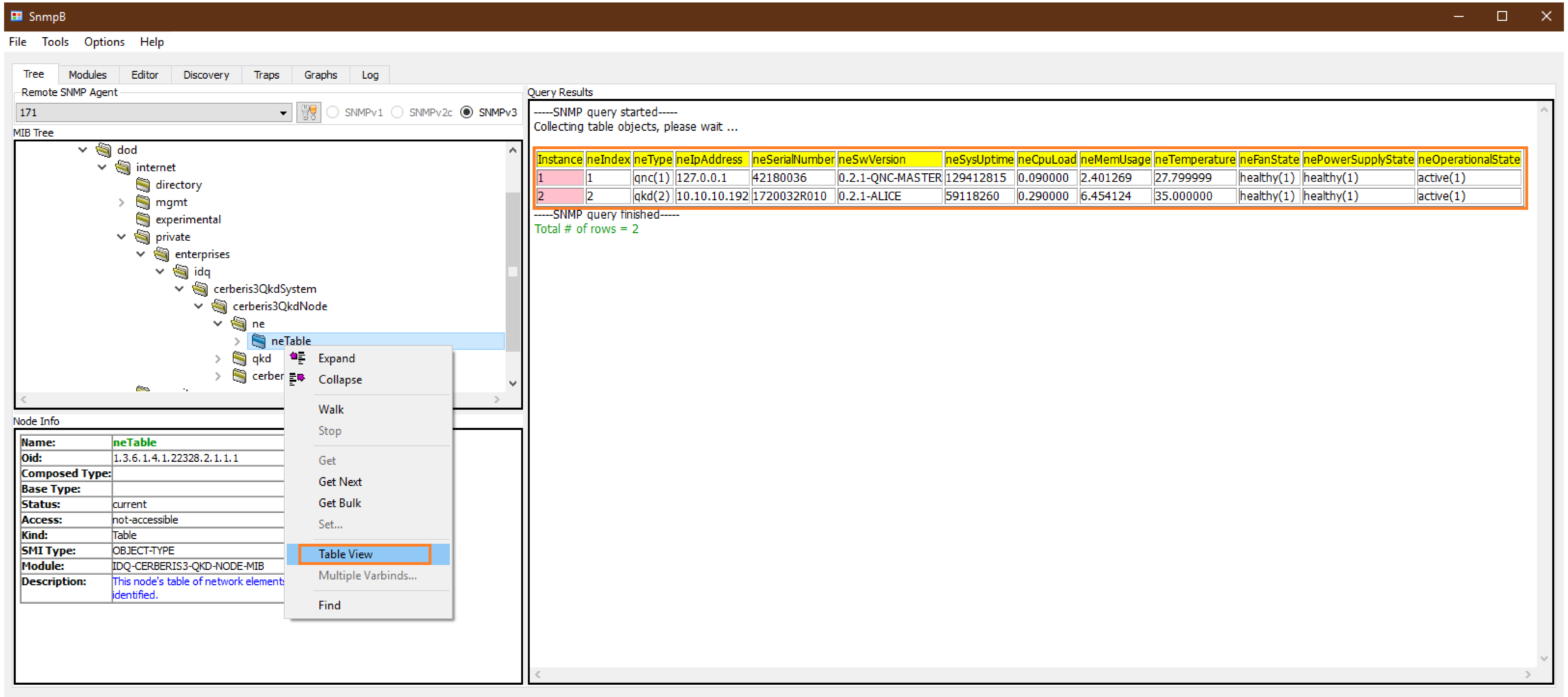
- Go to MIB Tree → iso → org → dod → internet → private → enterprise → idq → cerberis3QkdSystem



SNMPB configuration

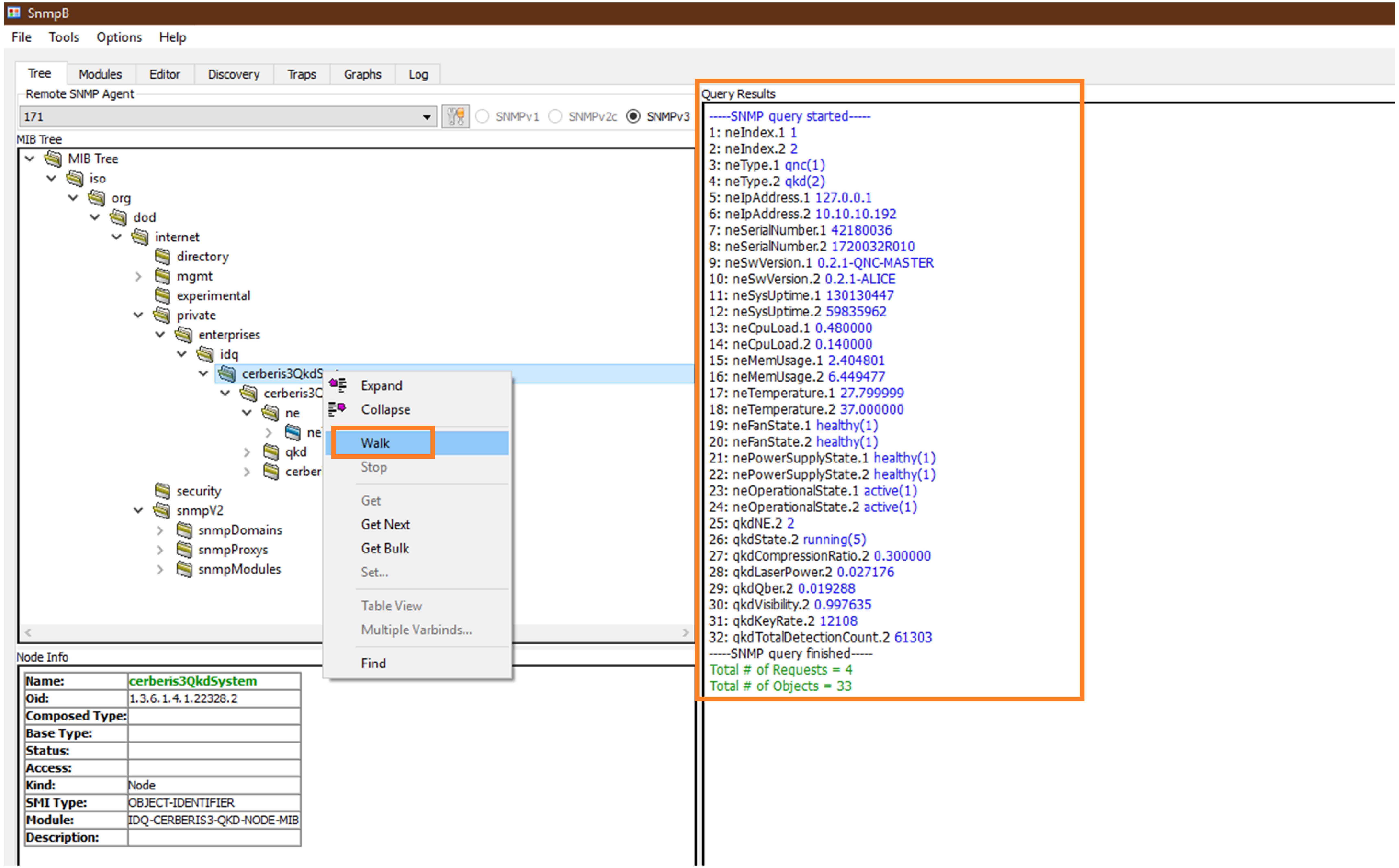
7. Check if the communication is working

- Expand NE and select “neTable”
- Do a right click on Table View and check if the values appear on the right side



SNMP Monitoring – Get values

- Expand “cerberis3QkdSystem”, do a right click and then press “Walk”



SNMP Monitoring – Get values

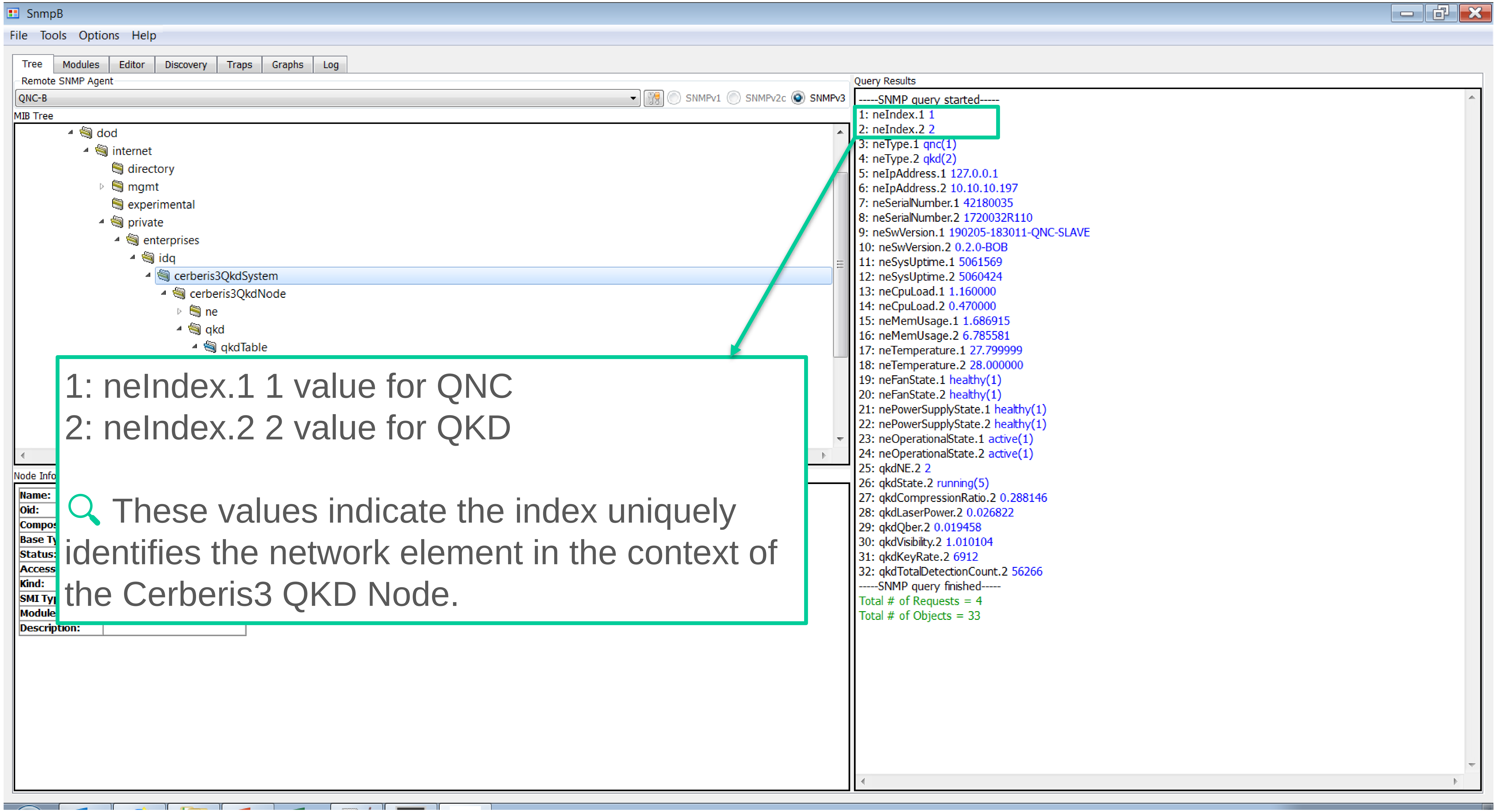
Cerberis3

- Values from QKD are transferred to QNC
- SNMP server is installed on the QNC
- You talk only with the QNC to request SNMP information.
- You don't have to talk with the QKD directly for SNMP requests.

CerberisXG

- **SNMP is hosted directly on the CerberisXG**

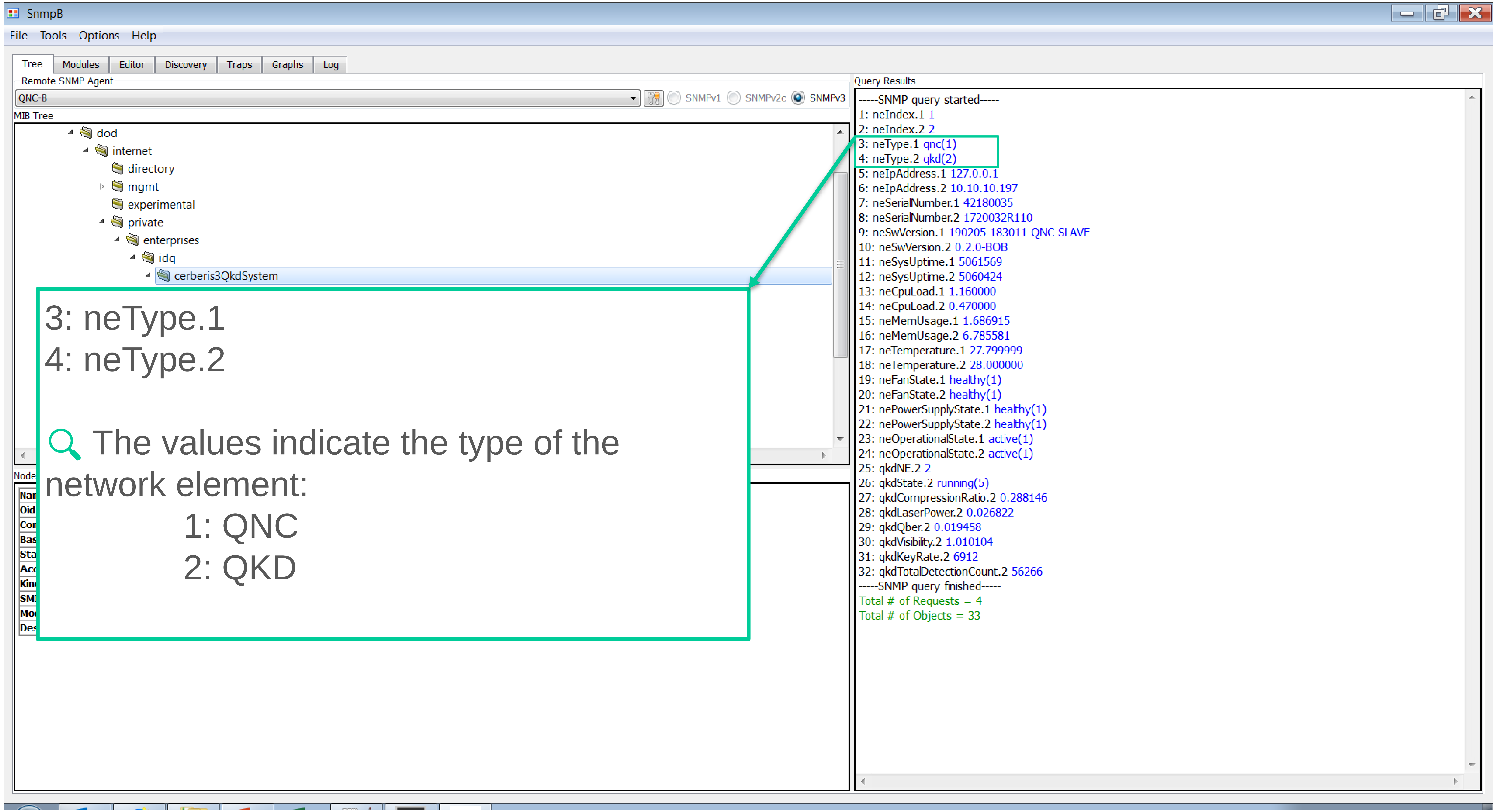
SNMP Monitoring – Values Explanations



1: neIndex.1 1 value for QNC
2: neIndex.2 2 value for QKD

These values indicate the index uniquely identifies the network element in the context of the Cerberis3 QKD Node.

SNMP Monitoring – Values Explanations

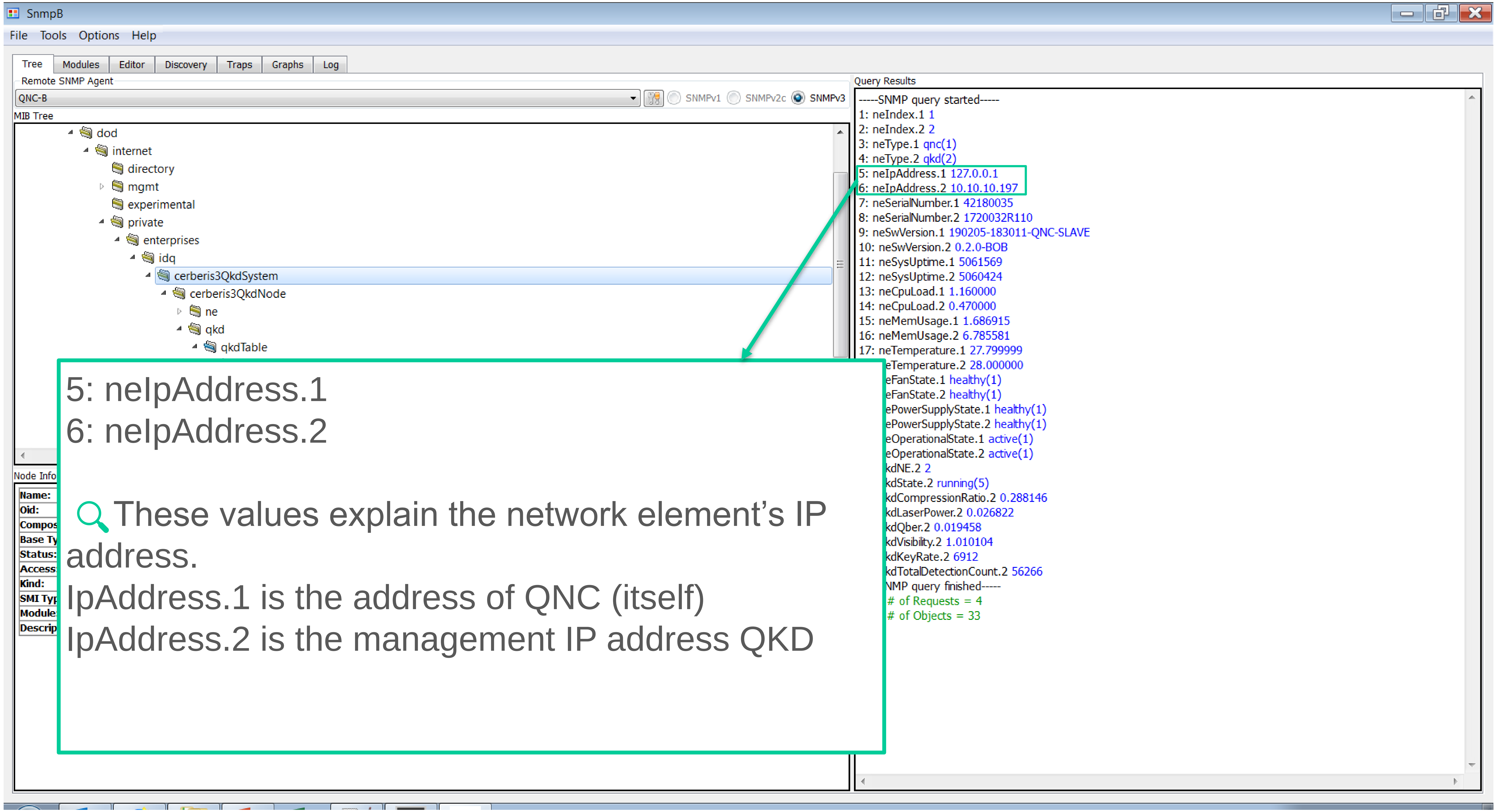


3: neType.1
4: neType.2

The values indicate the type of the network element:

1: QNC
2: QKD

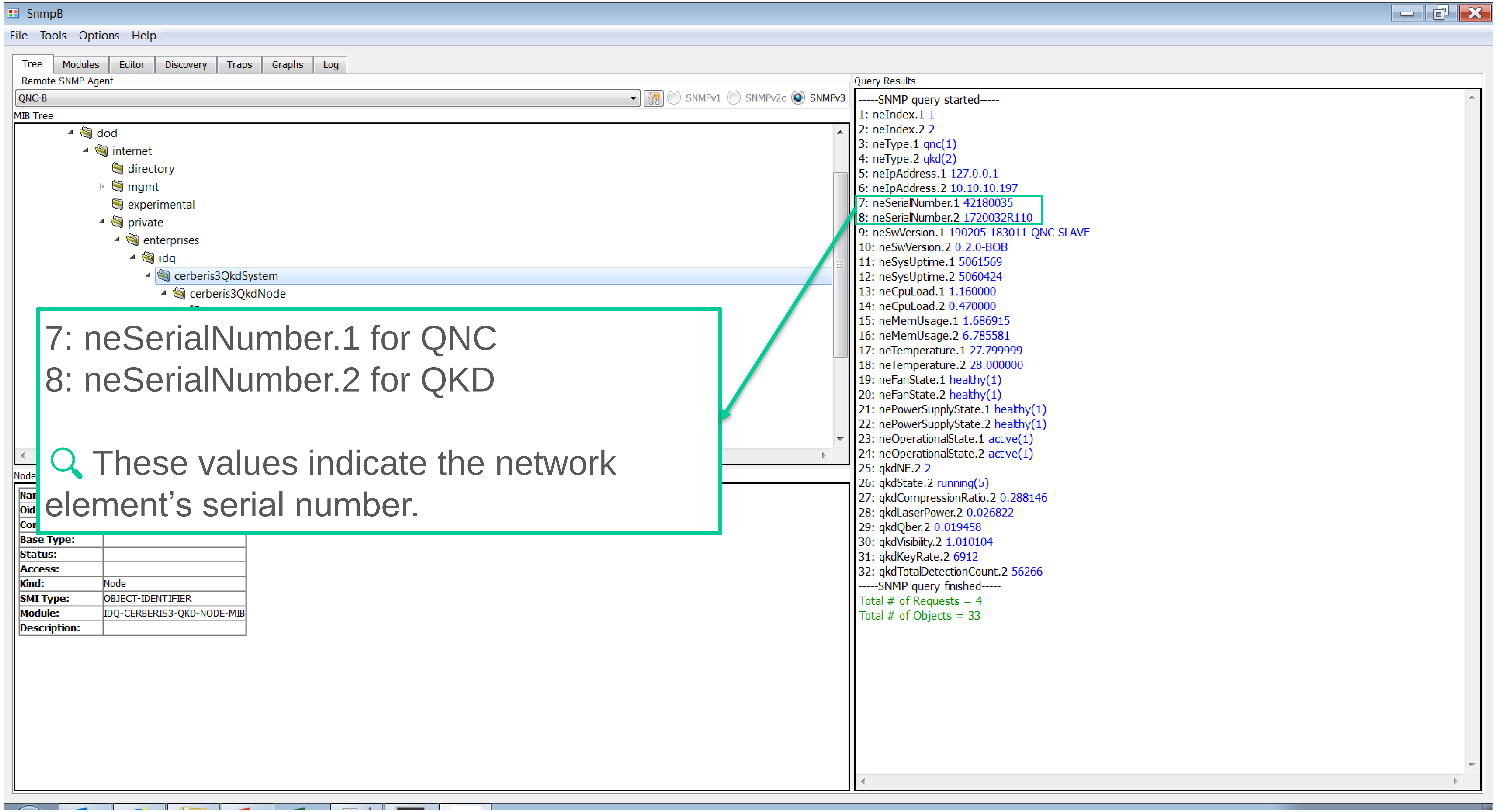
SNMP Monitoring – Values Explanations



5: neIpAddress.1
6: neIpAddress.2

These values explain the network element's IP address.
IpAddress.1 is the address of QNC (itself)
IpAddress.2 is the management IP address QKD

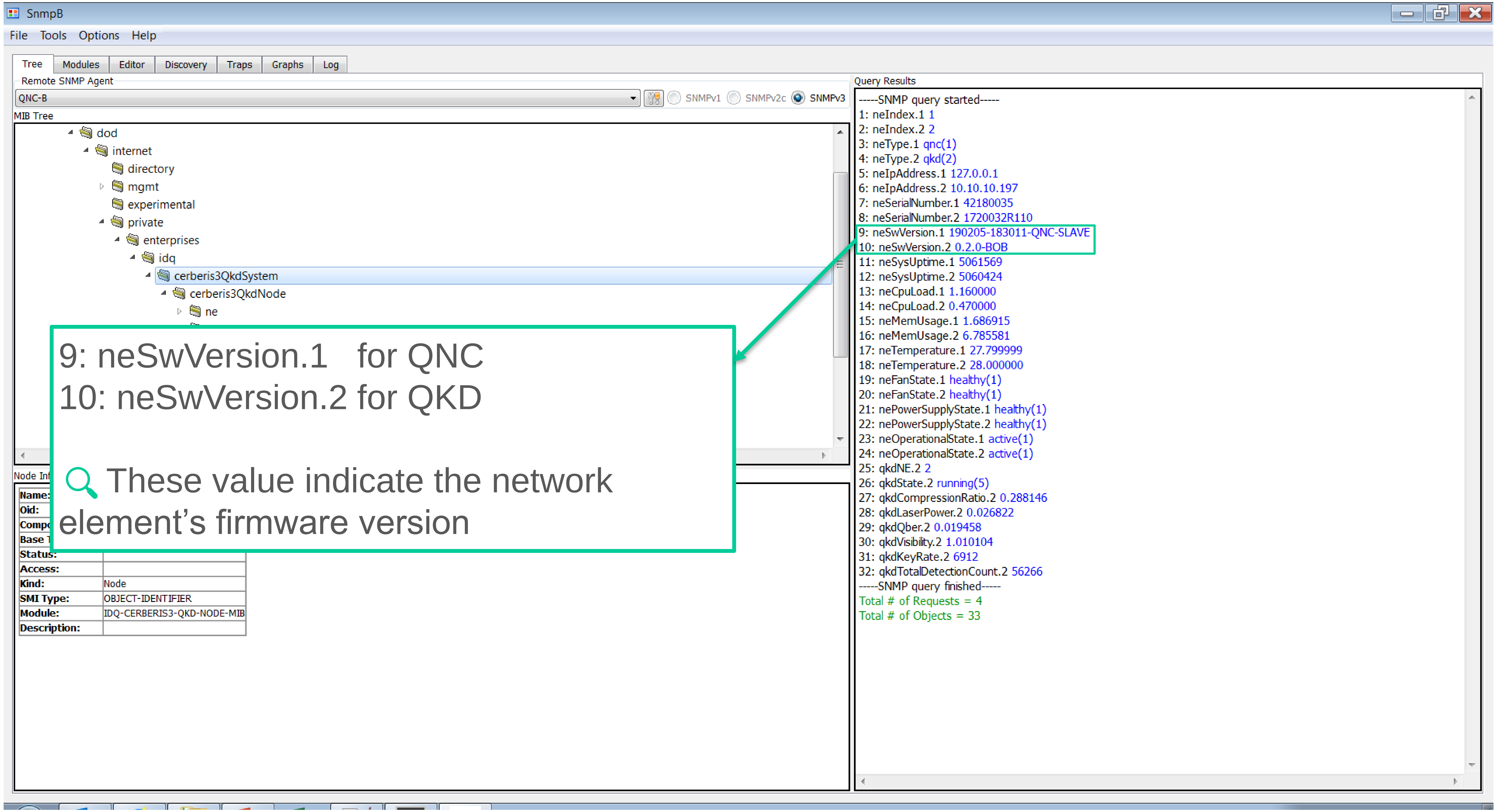
SNMP Monitoring – Values Explanations



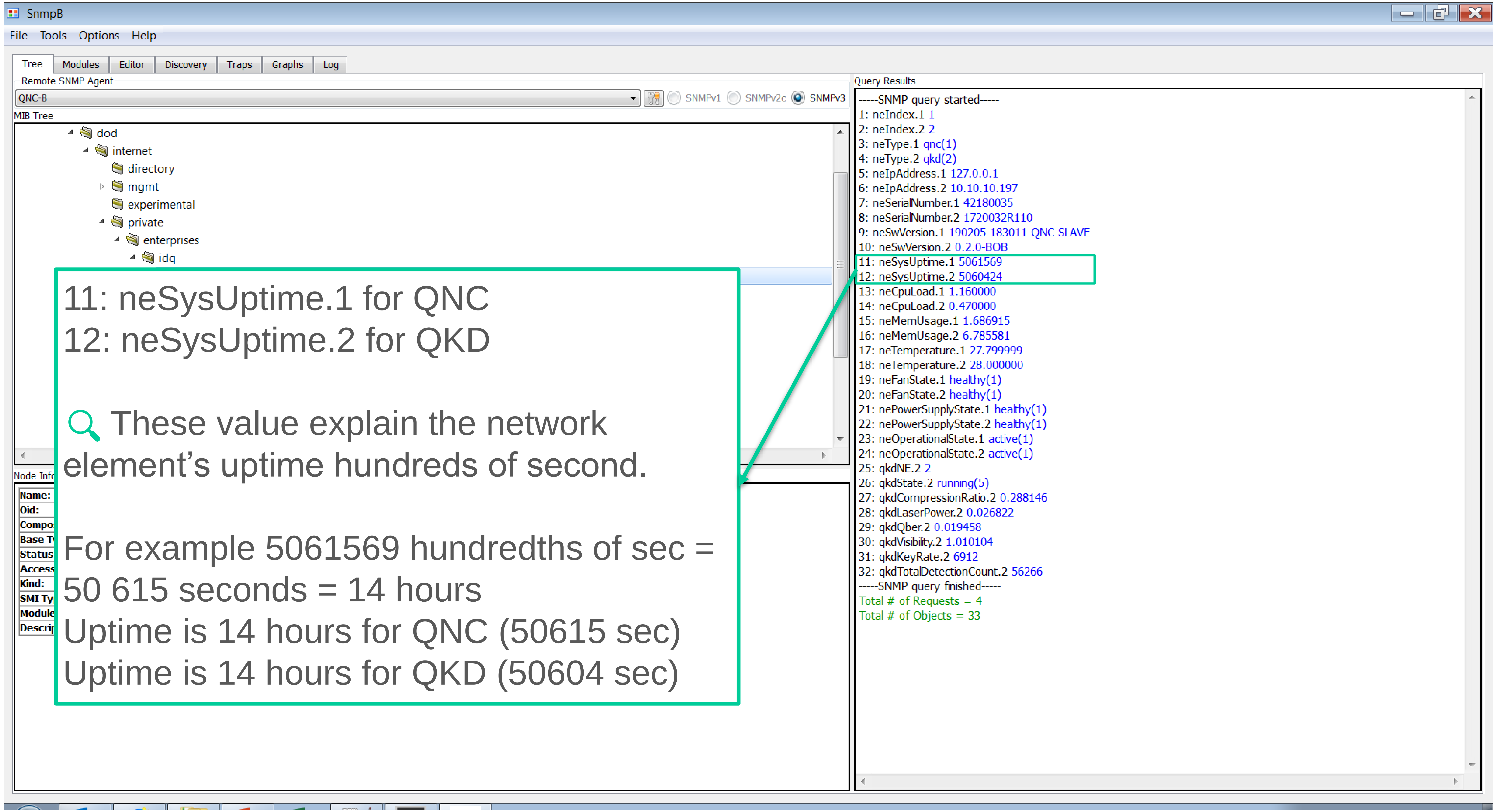
7: neSerialNumber.1 for QNC
8: neSerialNumber.2 for QKD

These values indicate the network element's serial number.

SNMP Monitoring – Values Explanations



SNMP Monitoring – Values Explanations

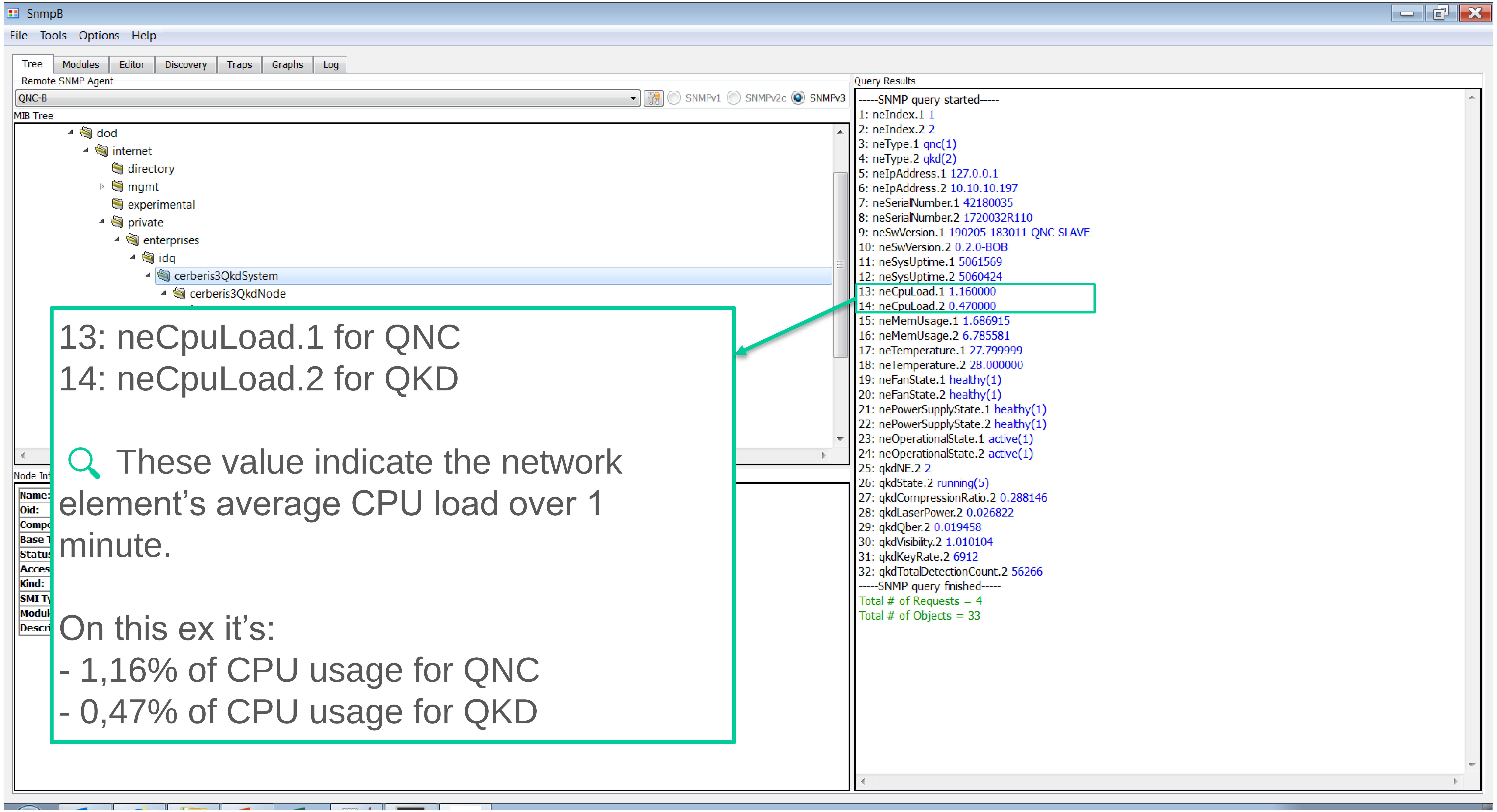


11: neSysUptime.1 for QNC
12: neSysUptime.2 for QKD

These value explain the network element's uptime hundreds of second.

For example 5061569 hundredths of sec = 50 615 seconds = 14 hours
Uptime is 14 hours for QNC (50615 sec)
Uptime is 14 hours for QKD (50604 sec)

SNMP Monitoring – Values Explanations

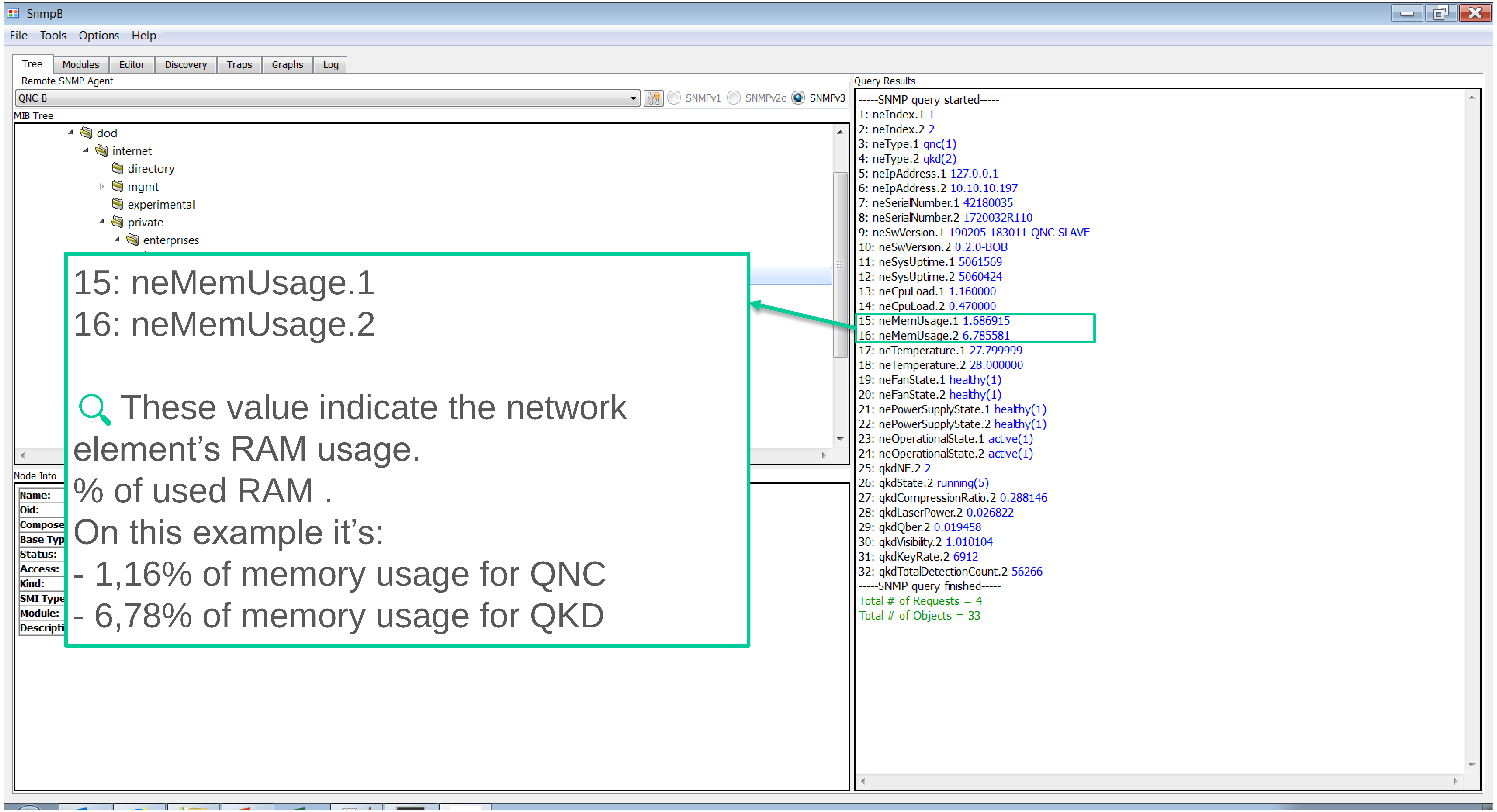


13: neCpuLoad.1 for QNC
14: neCpuLoad.2 for QKD

These value indicate the network element's average CPU load over 1 minute.

On this ex it's:
- 1,16% of CPU usage for QNC
- 0,47% of CPU usage for QKD

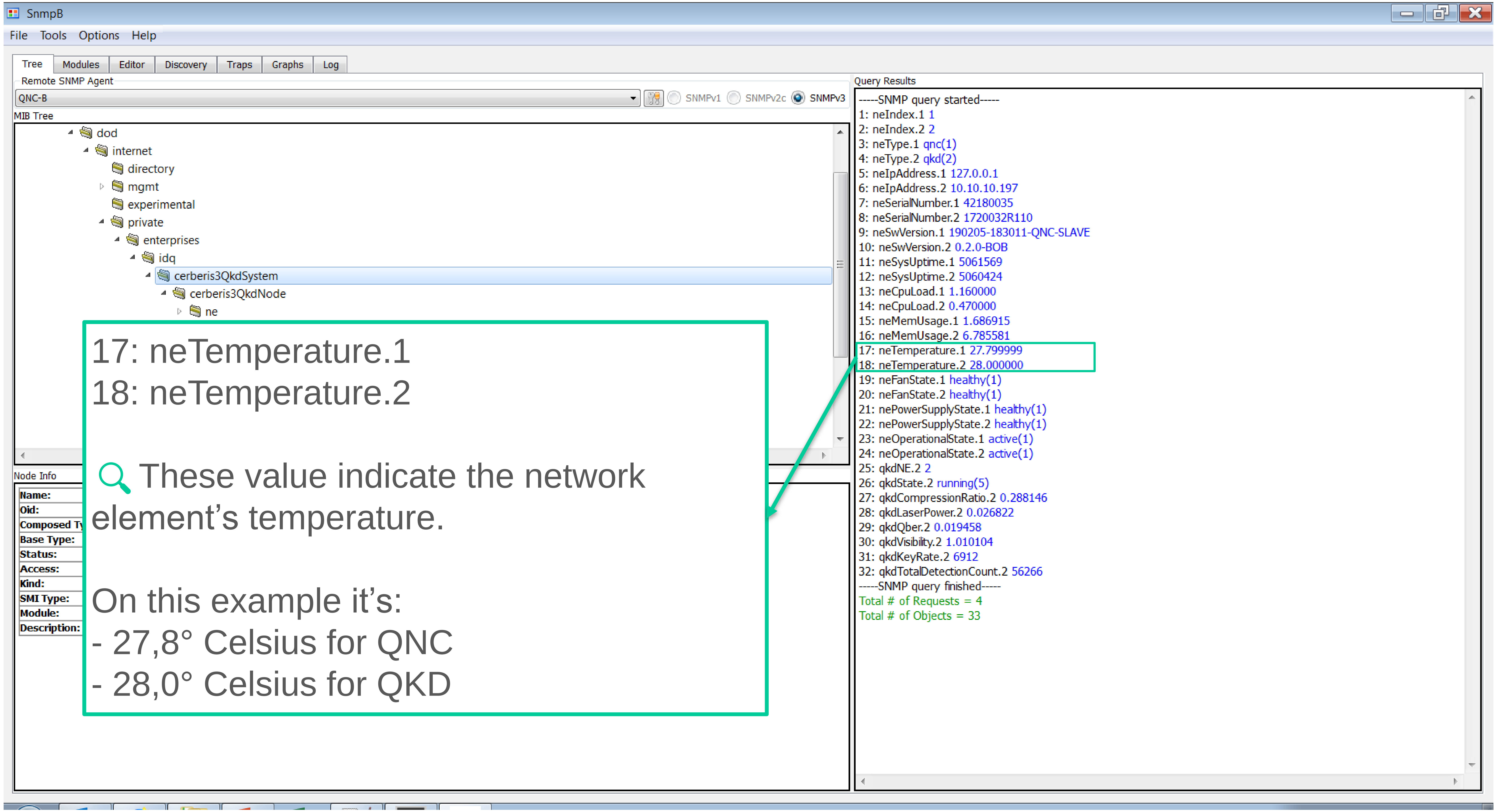
SNMP Monitoring – Values Explanations



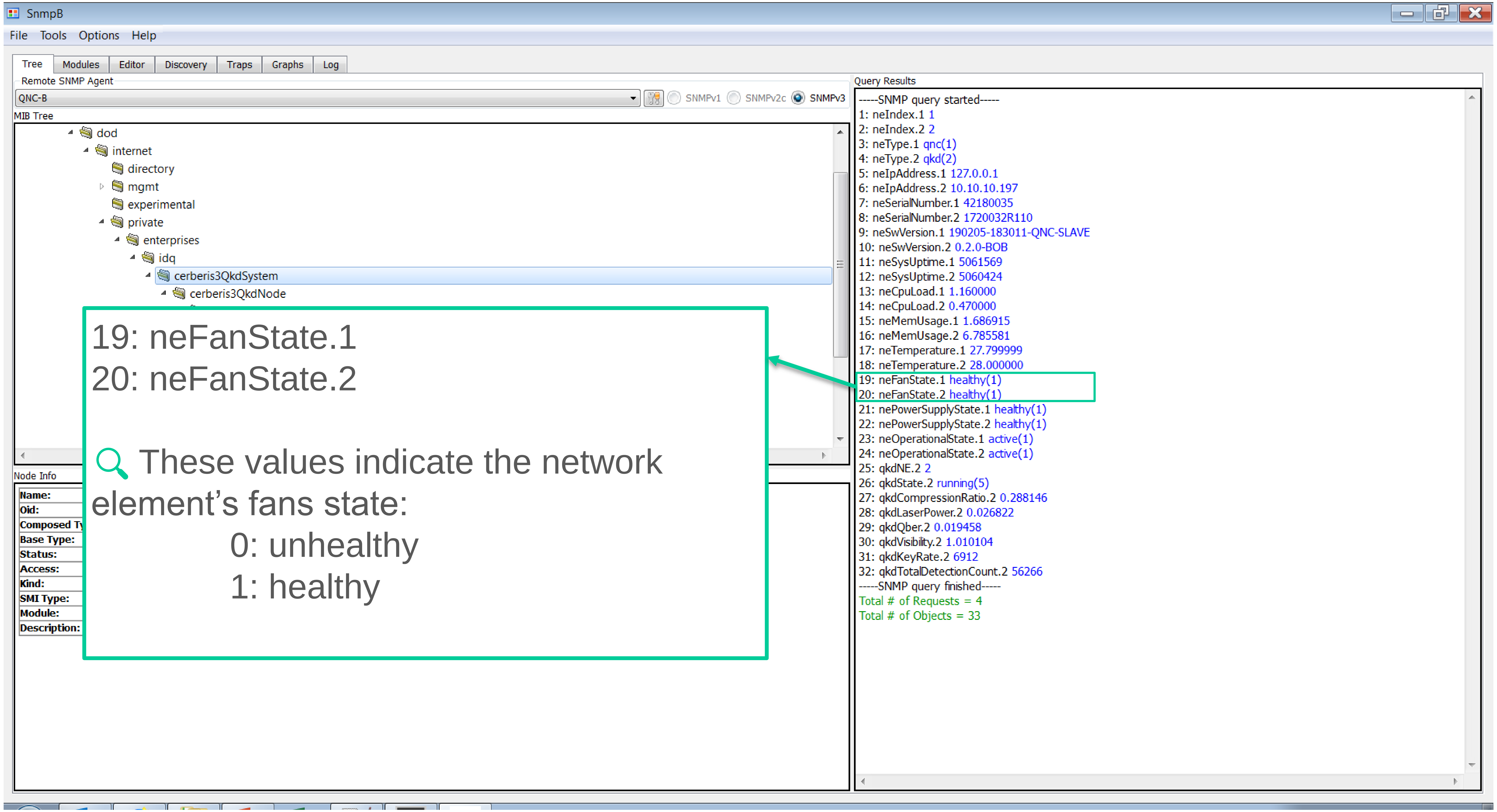
15: neMemUsage.1
16: neMemUsage.2

These value indicate the network element's RAM usage.
% of used RAM .
On this example it's:
- 1,16% of memory usage for QNC
- 6,78% of memory usage for QKD

SNMP Monitoring – Values Explanations



SNMP Monitoring – Values Explanations

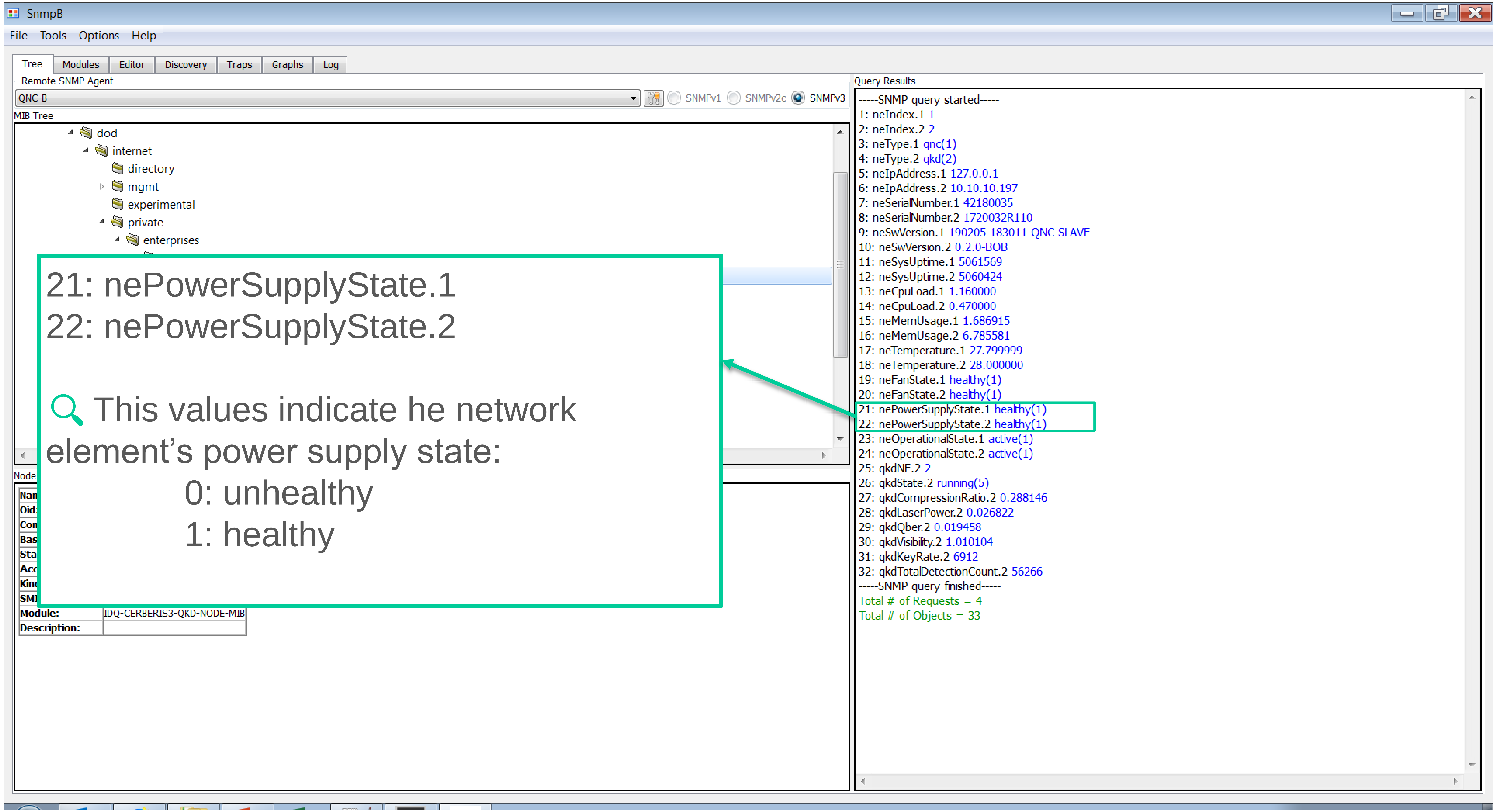


19: neFanState.1
20: neFanState.2

These values indicate the network element's fans state:

0: unhealthy
1: healthy

SNMP Monitoring – Values Explanations

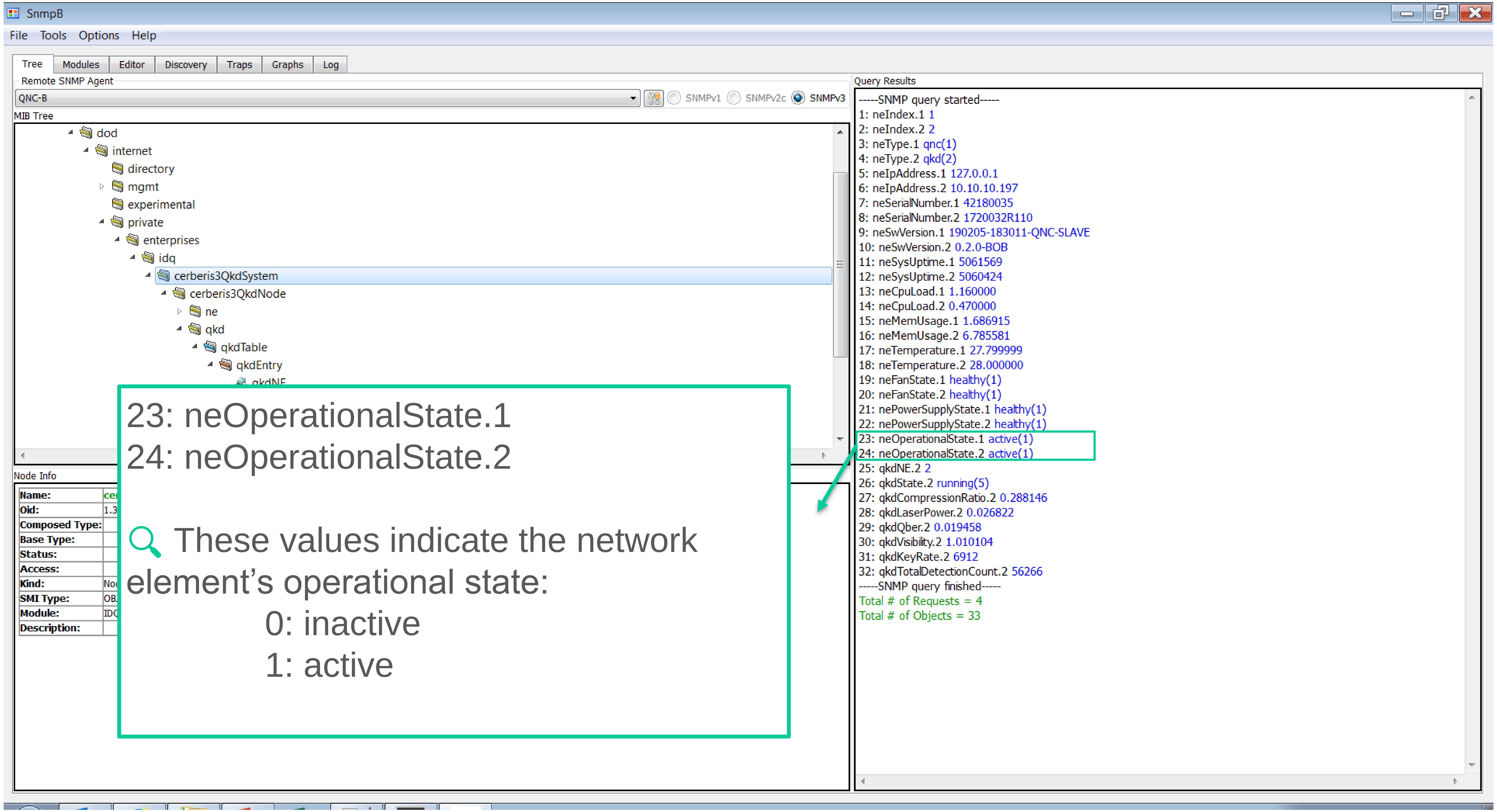


21: nePowerSupplyState.1
22: nePowerSupplyState.2

This values indicate he network element's power supply state:

0: unhealthy
1: healthy

SNMP Monitoring – Values Explanations



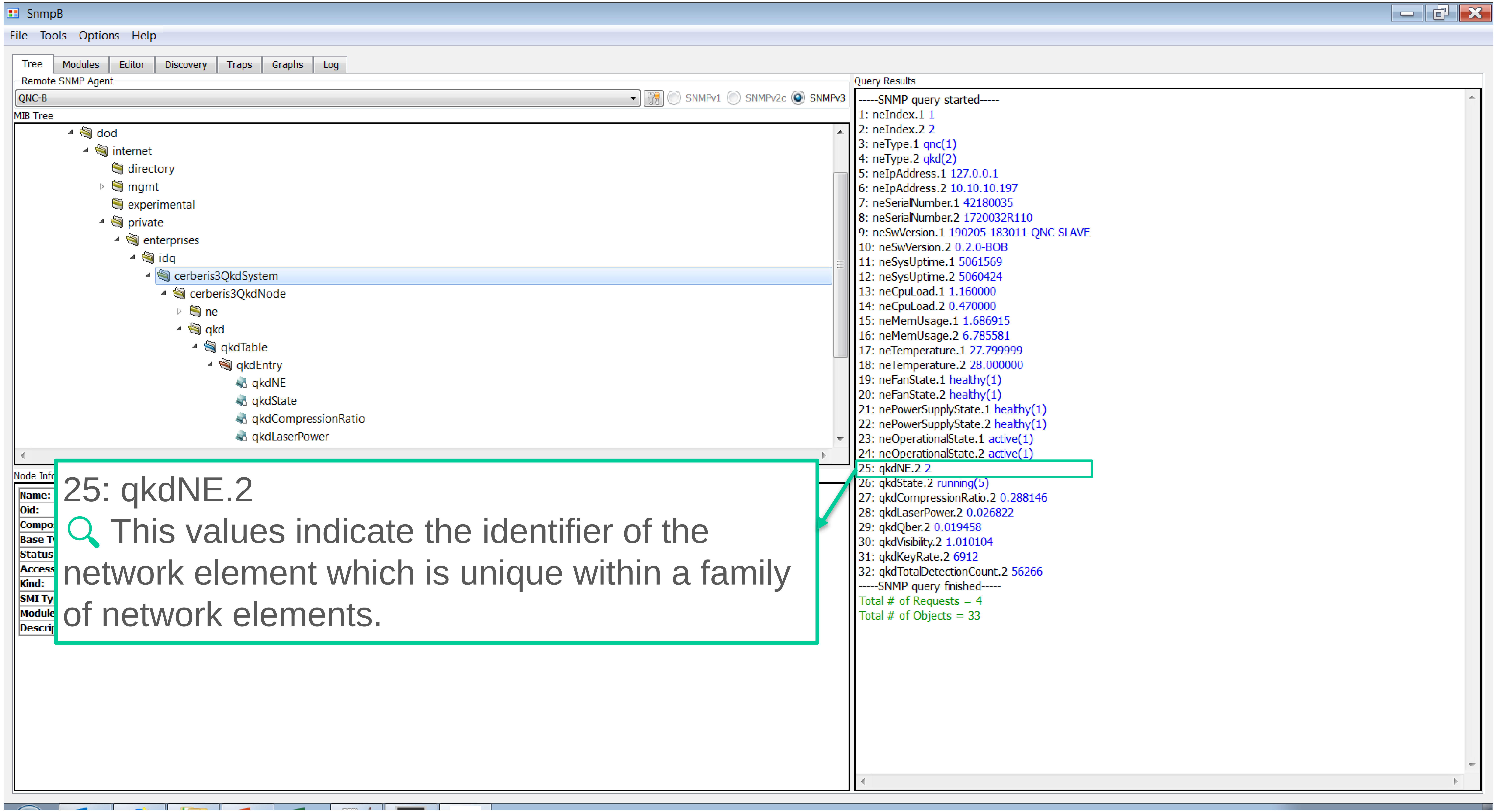
23: neOperationalState.1
24: neOperationalState.2

These values indicate the network element's operational state:

0: inactive
1: active

SNMP Monitoring – Values Explanations

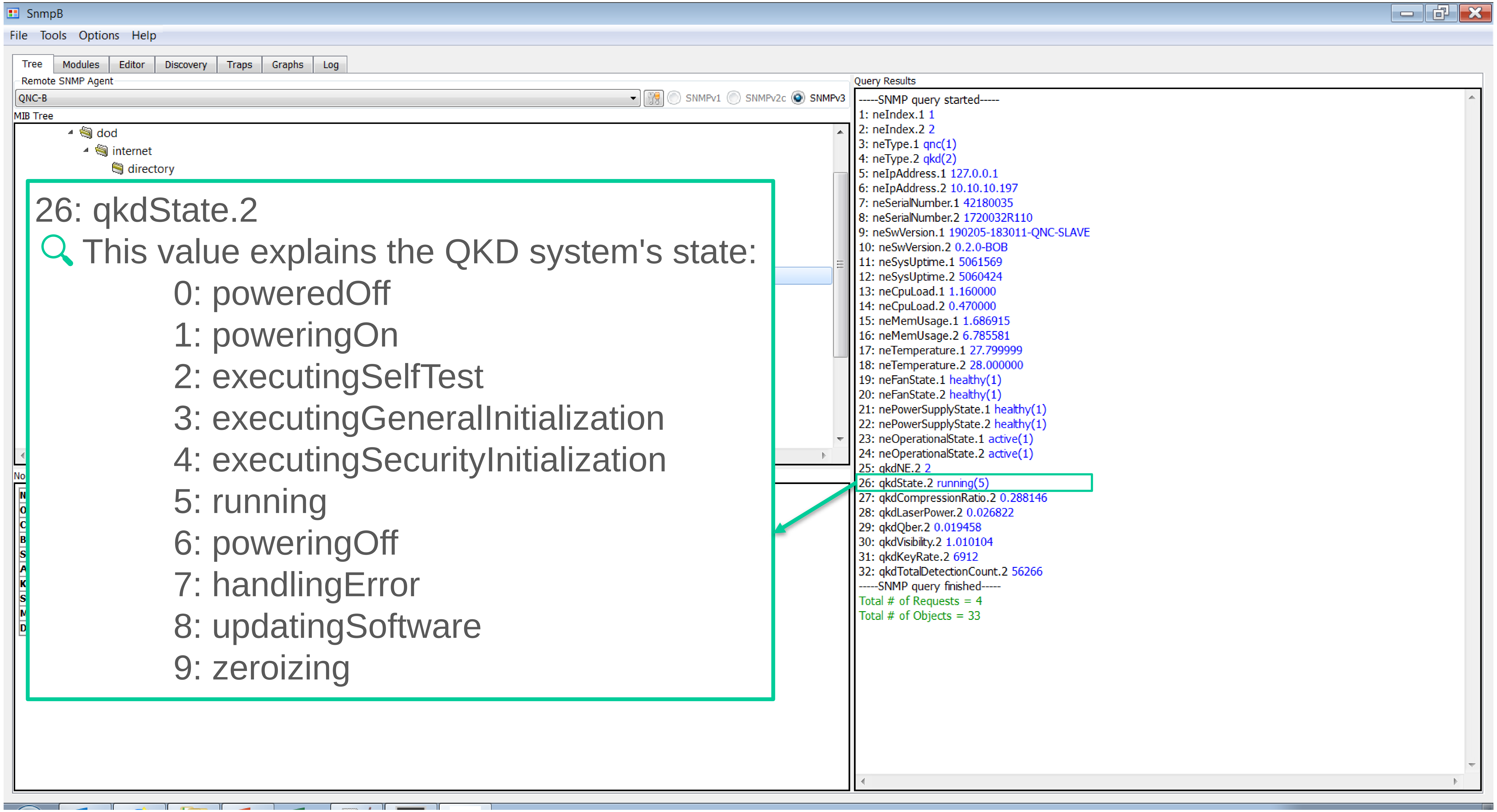
SWISS QUANTUM



25: qkdNE.2

This values indicate the identifier of the network element which is unique within a family of network elements.

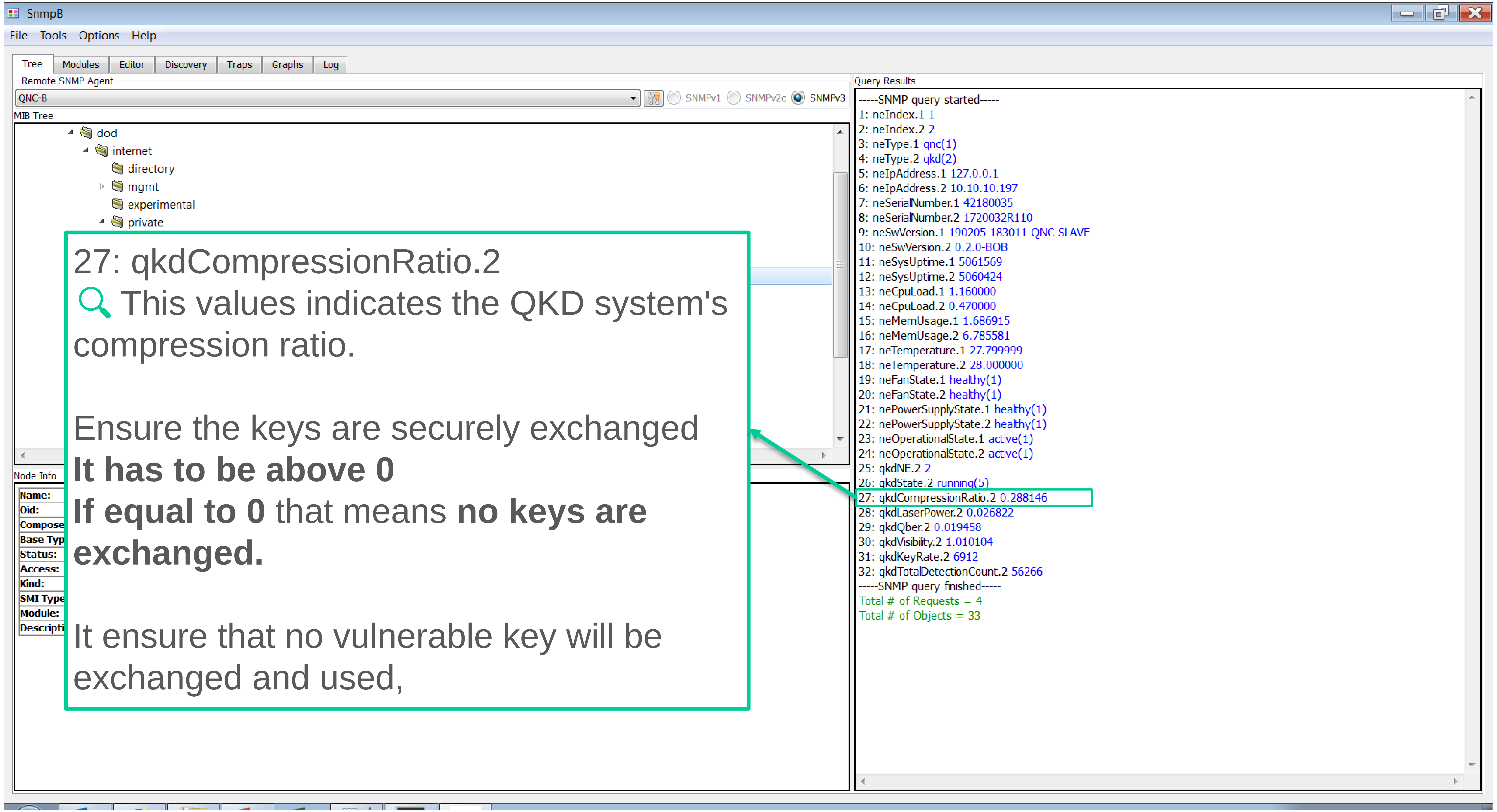
SNMP Monitoring – Values Explanations



SNMP Monitoring – Values Explanations

0: poweredOff
→ Communicator's service is stopped
1: poweringOn
→ Communicator's service is starting
2: executingSelfTest SelfTest
→ A series of tests is processing when the service is starting. This stage could take up to 15 minutes because the system Bob needs to reach a low temperature.
3: executingGeneralInitialization
→ This is the alignment procedure. It can take up to 15 minutes
4: executingSecurityInitialization
→ That means the initial keys are set on QKD
5: running
→ The QKD processes are working. However that doesn't ensure that the QKD Keys are exchanged
6: poweringOff
→ Communicator's service is restarting because watchdog detected something wrong or user request a restart.
7: handlingError
→ When something is wrong. The system will restart the service or reboot the whole device
8: updatingSoftware
→ This is the needed condition when upgrading
9: zeroizing
→ This is the erasing procedure when tamper is detected

SNMP Monitoring – Values Explanations



27: qkdCompressionRatio.2

This values indicates the QKD system's compression ratio.

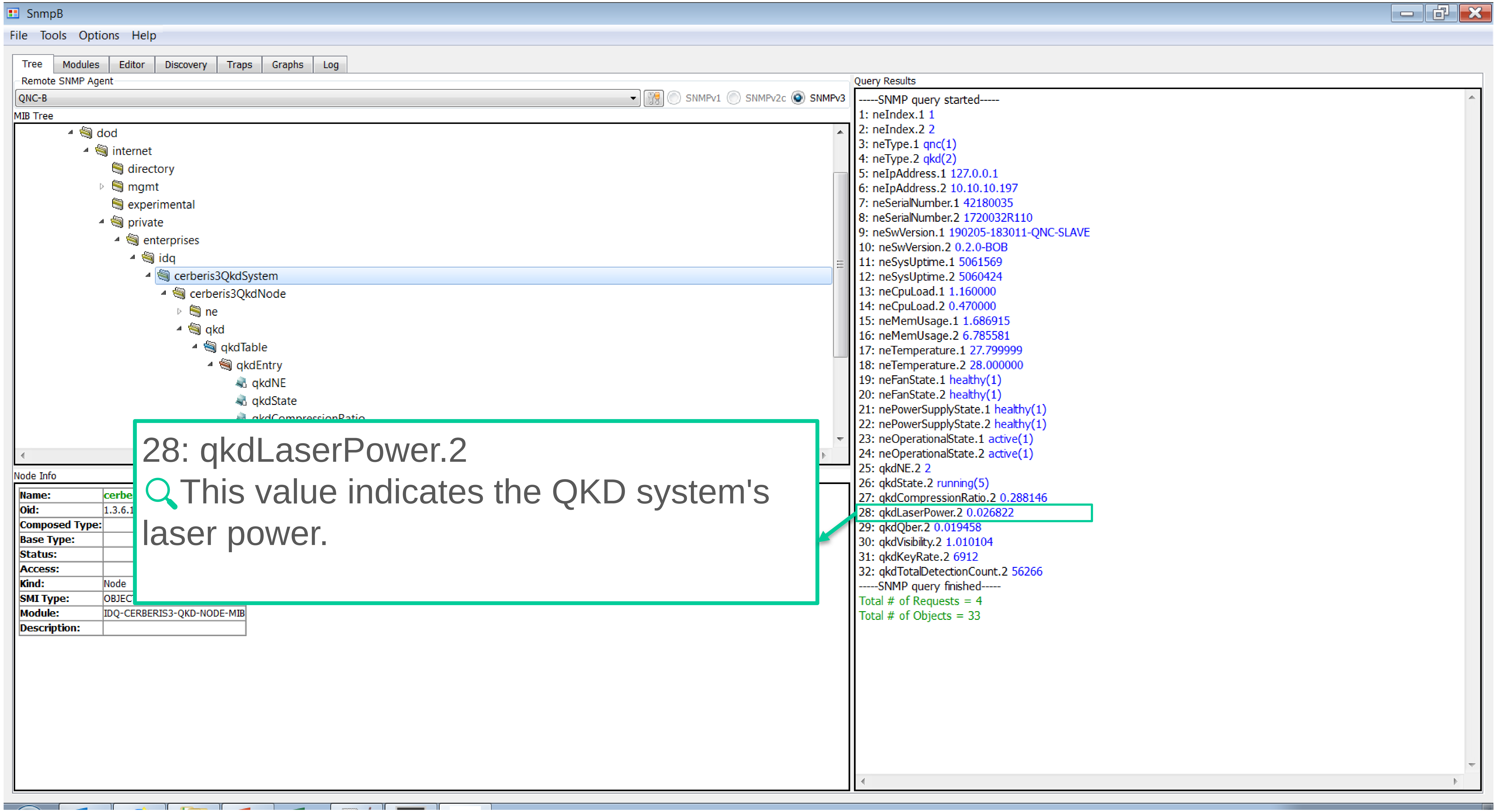
Ensure the keys are securely exchanged

It has to be above 0

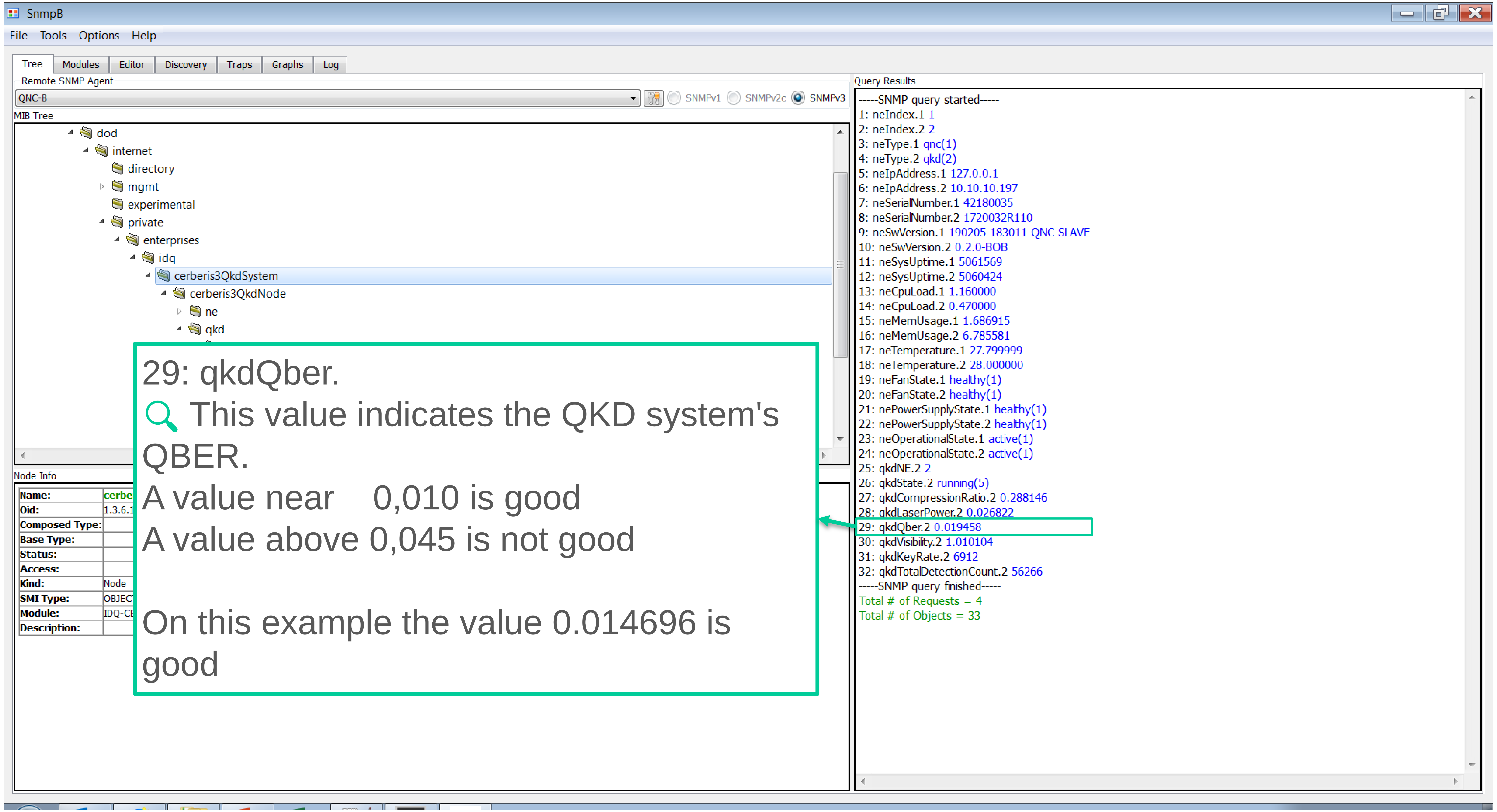
If equal to 0 that means **no keys are exchanged.**

It ensure that no vulnerable key will be exchanged and used,

SNMP Monitoring – Values Explanations



SNMP Monitoring – Values Explanations



29: qkdQber.

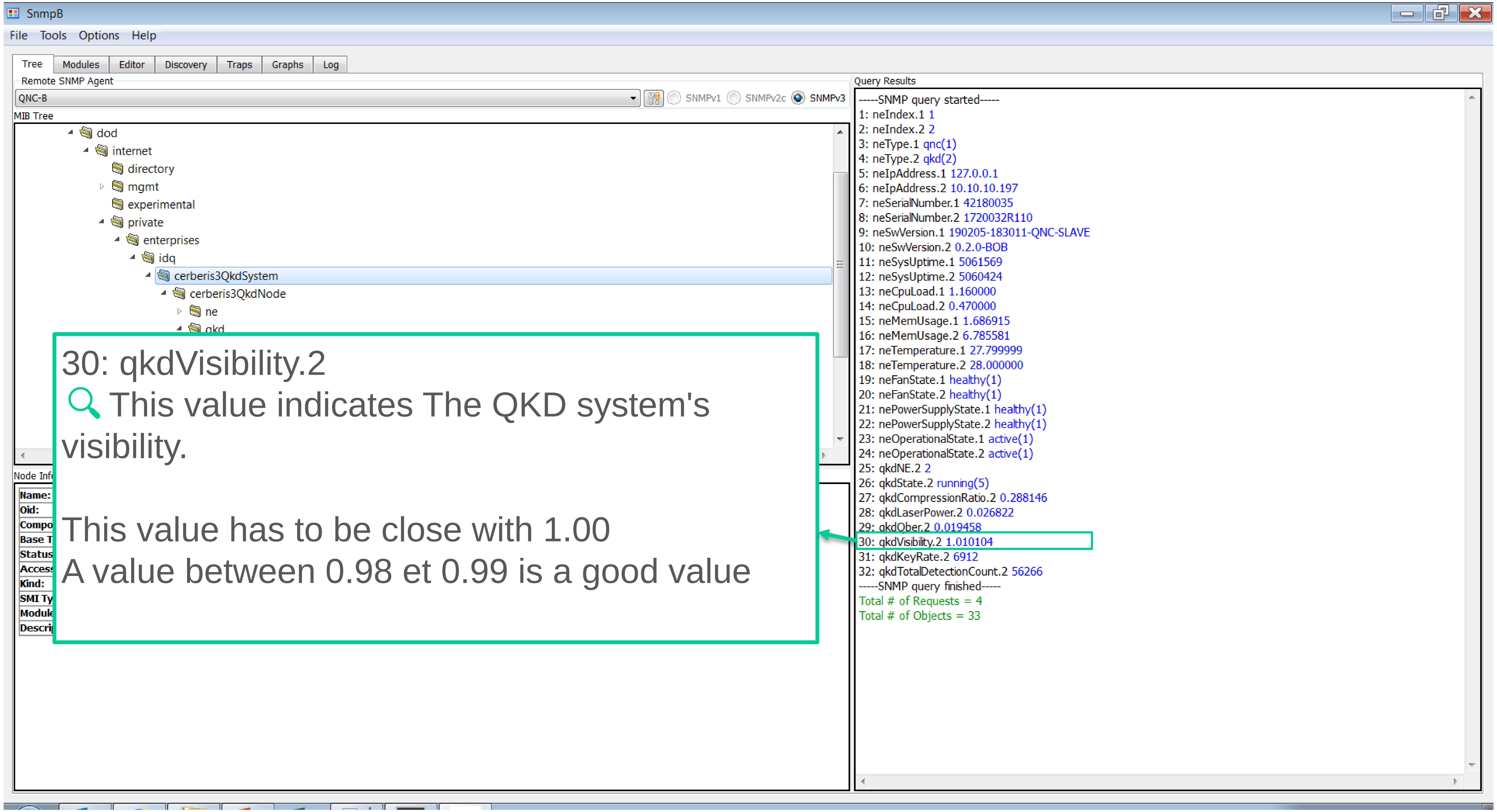
This value indicates the QKD system's QBER.

A value near 0,010 is good

A value above 0,045 is not good

On this example the value 0.014696 is good

SNMP Monitoring – Values Explanations



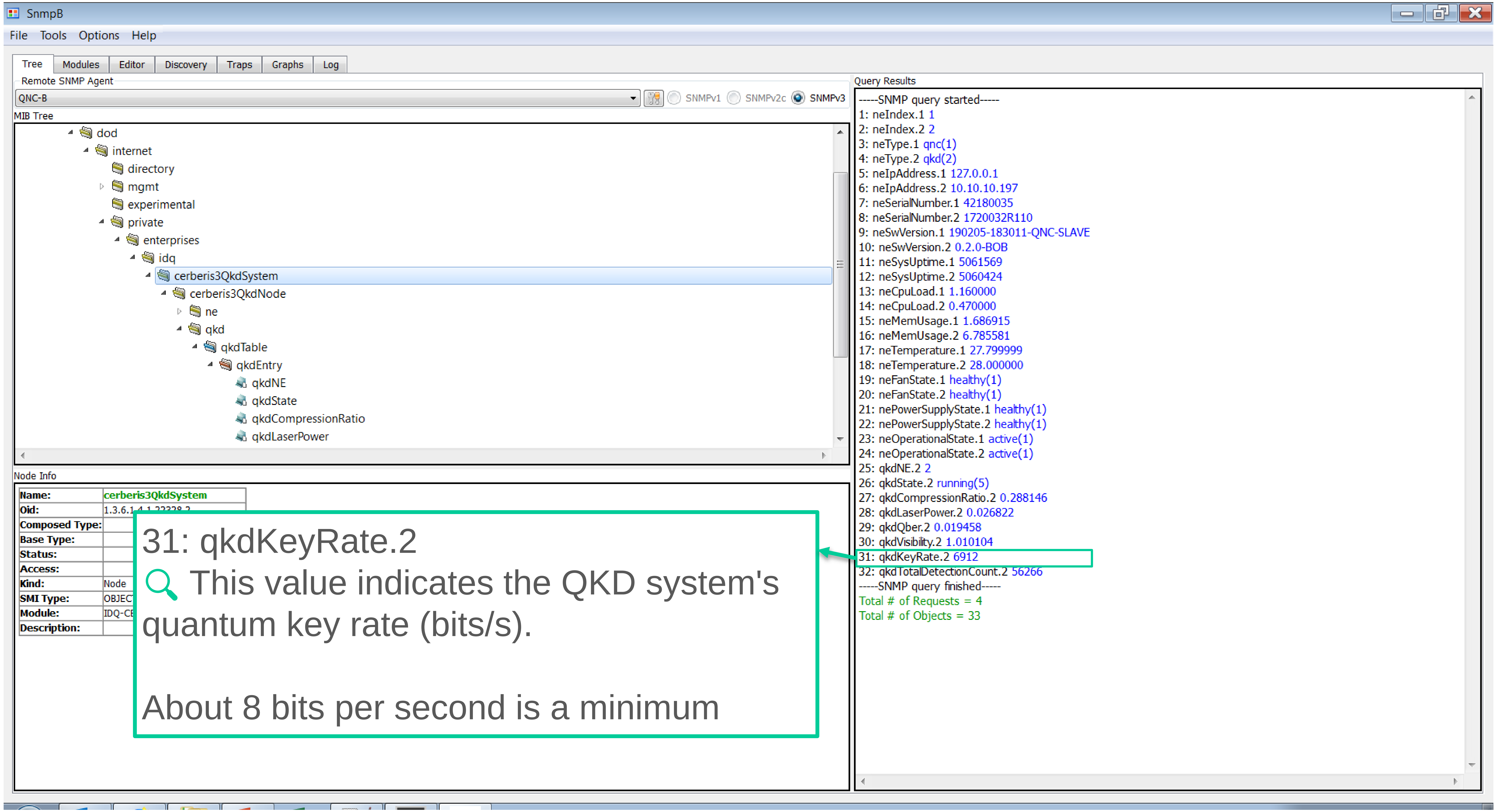
30: qkdVisibility.2

This value indicates The QKD system's visibility.

This value has to be close with 1.00

A value between 0.98 et 0.99 is a good value

SNMP Monitoring – Values Explanations

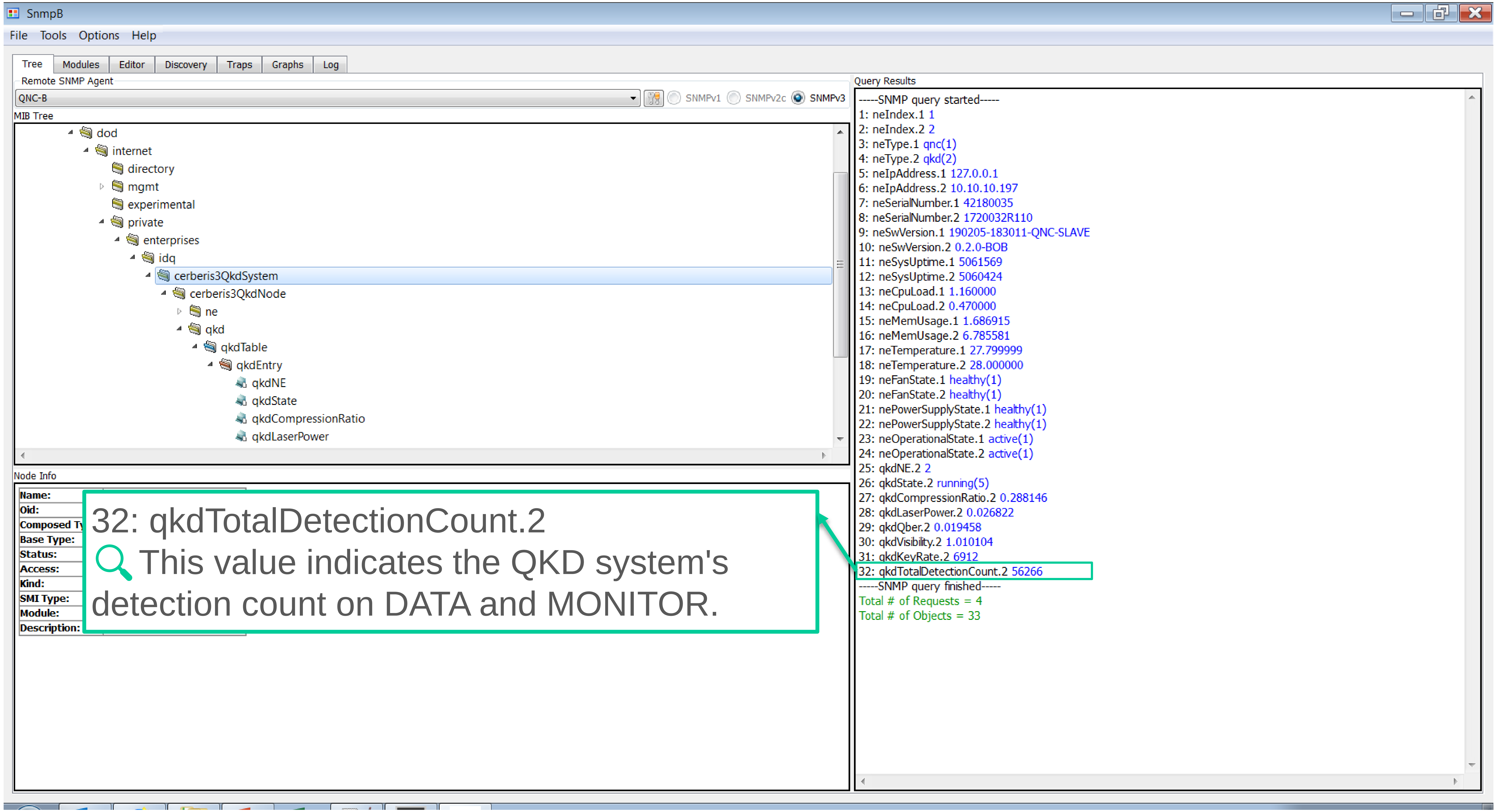


31: qkdKeyRate.2

This value indicates the QKD system's quantum key rate (bits/s).

About 8 bits per second is a minimum

SNMP Monitoring – Values Explanations



32: qkdTotalDetectionCount.2

This value indicates the QKD system's detection count on DATA and MONITOR.

SNMP Monitoring – Summarize

Summarize

- 2 values **are important to check:**

  - **Visibility** (it has to be close with 1)
  - **QBER** (approx 2-3 %) (4.5% max)
  - **Key Rate** (minimum 8 bits per second)

- If these two values are correct the QKDs **should be able to deliver keys to encryptors**

- Key delivery should be checked on the encryptors

ETSI client

ETSI client

SWISS QUANTUM

- **ETSI** is the primary protocol to connect third party devices to the CerberisXG or QNCs (We support others such as Cisco, etc)

- Usually the third party devices are:
  - **Encryptors**
  - **HSM**
  - Encryptor **simulators**

- **Regardless** of the type of third party devices, the processes are quite similar

- **ETSI** will provide 256 bits keys based on Quantum Physics to third party devices

ETSI client

You have **2 choices**:

- Connect your ETSI client to the CerberisXG KEY interface/QNC frontpane interface

- Connect your ETSI client to the MGT/management interface

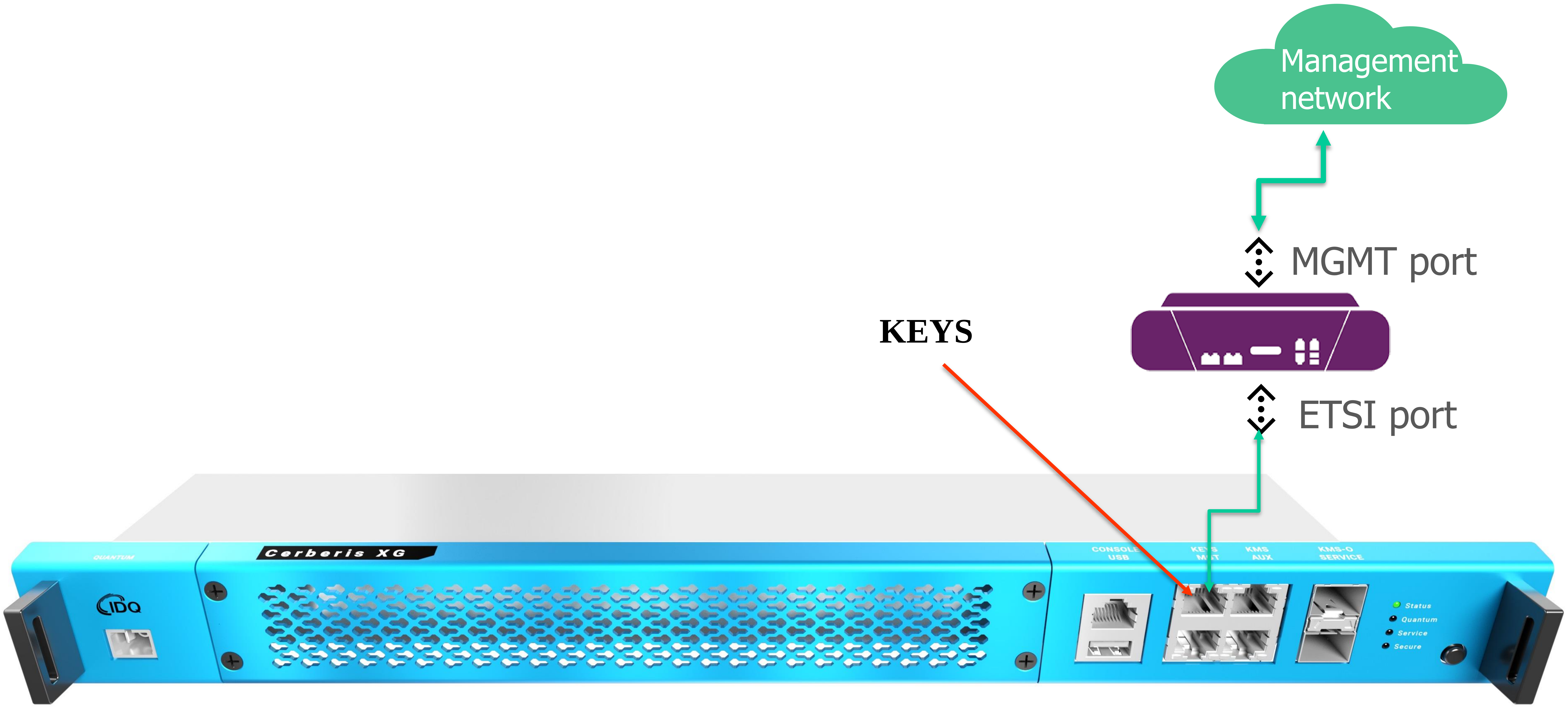
ETSI client

You have 2 choices:

- **Connect your ETSI client to the CerberisXG KEY interface/QNC frontpane interface**

- Connect your ETSI client to the MGT/management interface

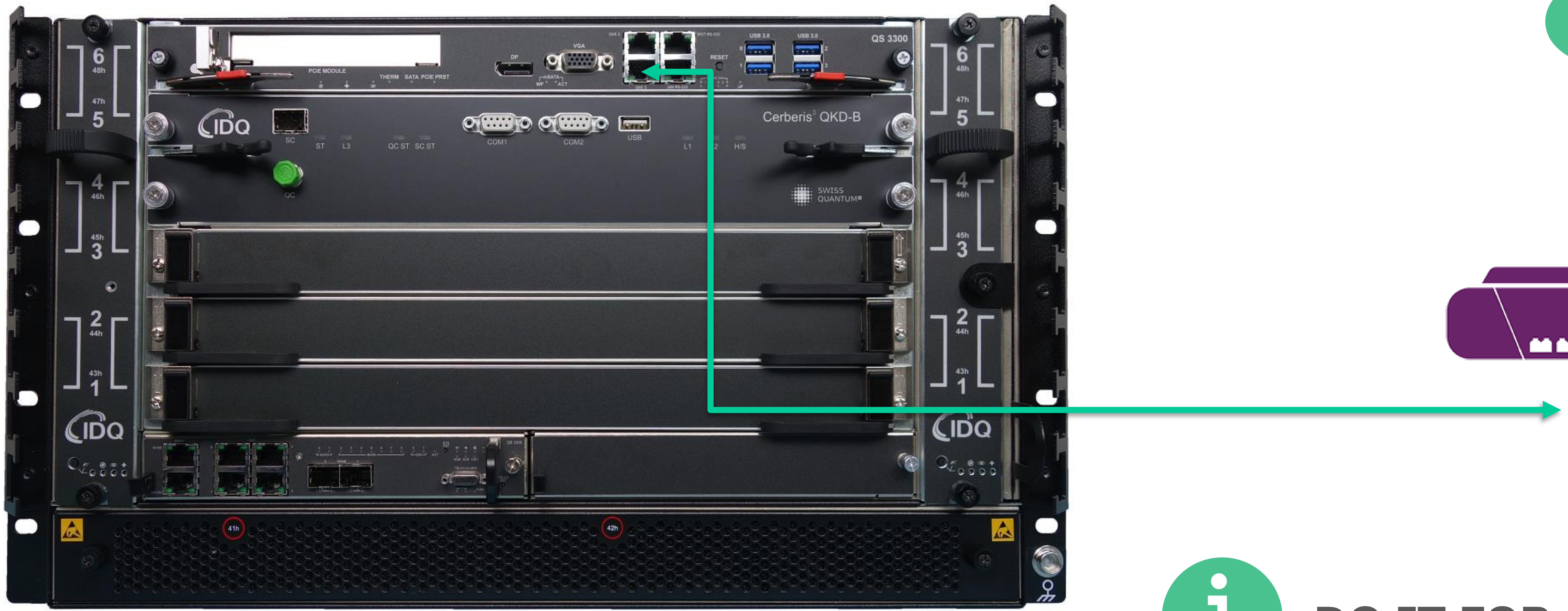
ETSI Cient

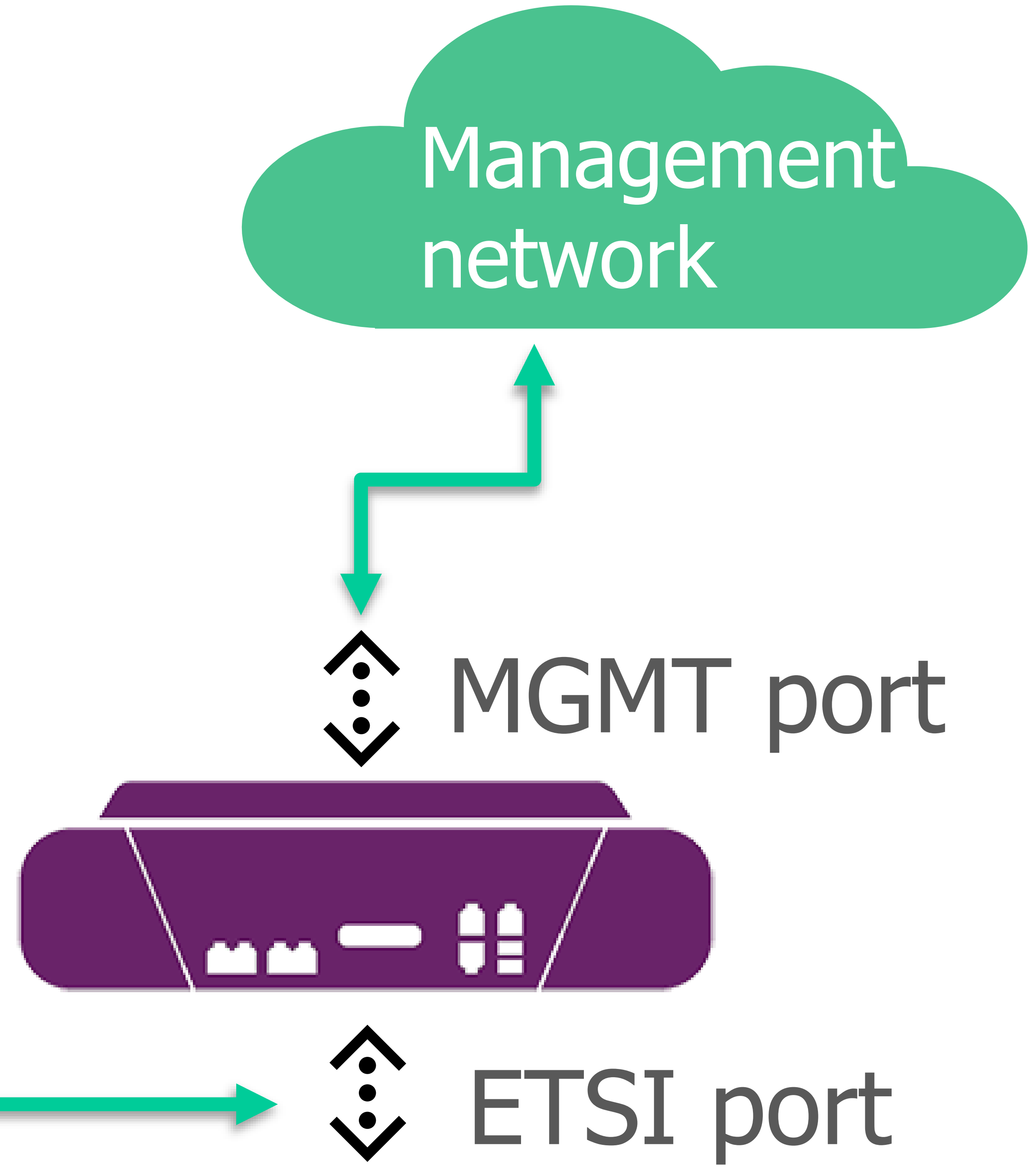

Management network
MGMT port
KEYS
ETSI port
Cerberis XG
CONSOLE USB
KMS AUX
KMS-Q SERVICE
Status
Quantum
Service
Secure

ETSI client

This topology concerns encryptors that have **2 ports**: management + dedicated ETSI client

→ **Connect directly** frontpane port to ETSI client port (encryptor)





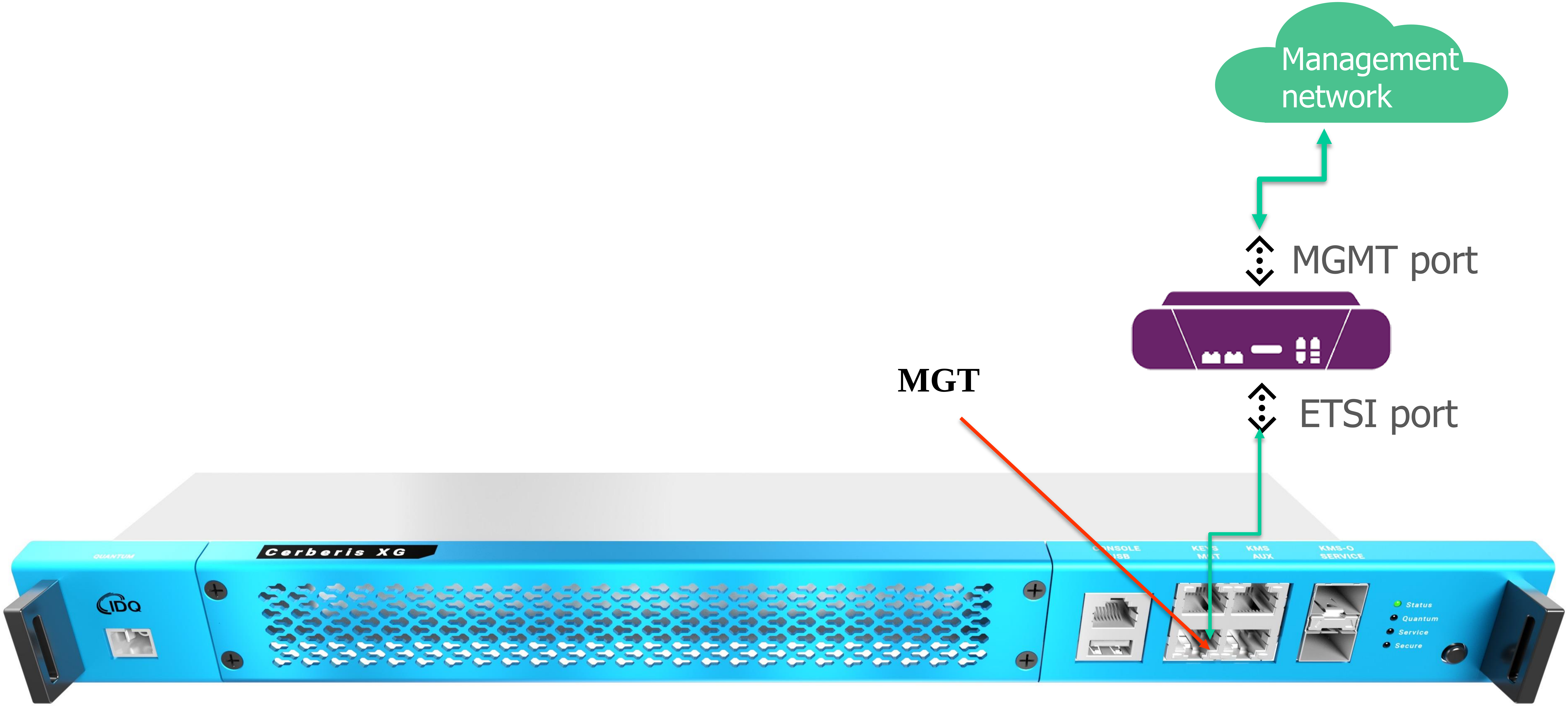
DO IT FOR EACH QNC

ETSI client

You have 2 choices:

- Connect your ETSI client to the CerberisXG KEY interface/QNC frontpane interface
- **Connect your ETSI client to the MGT/management interface**

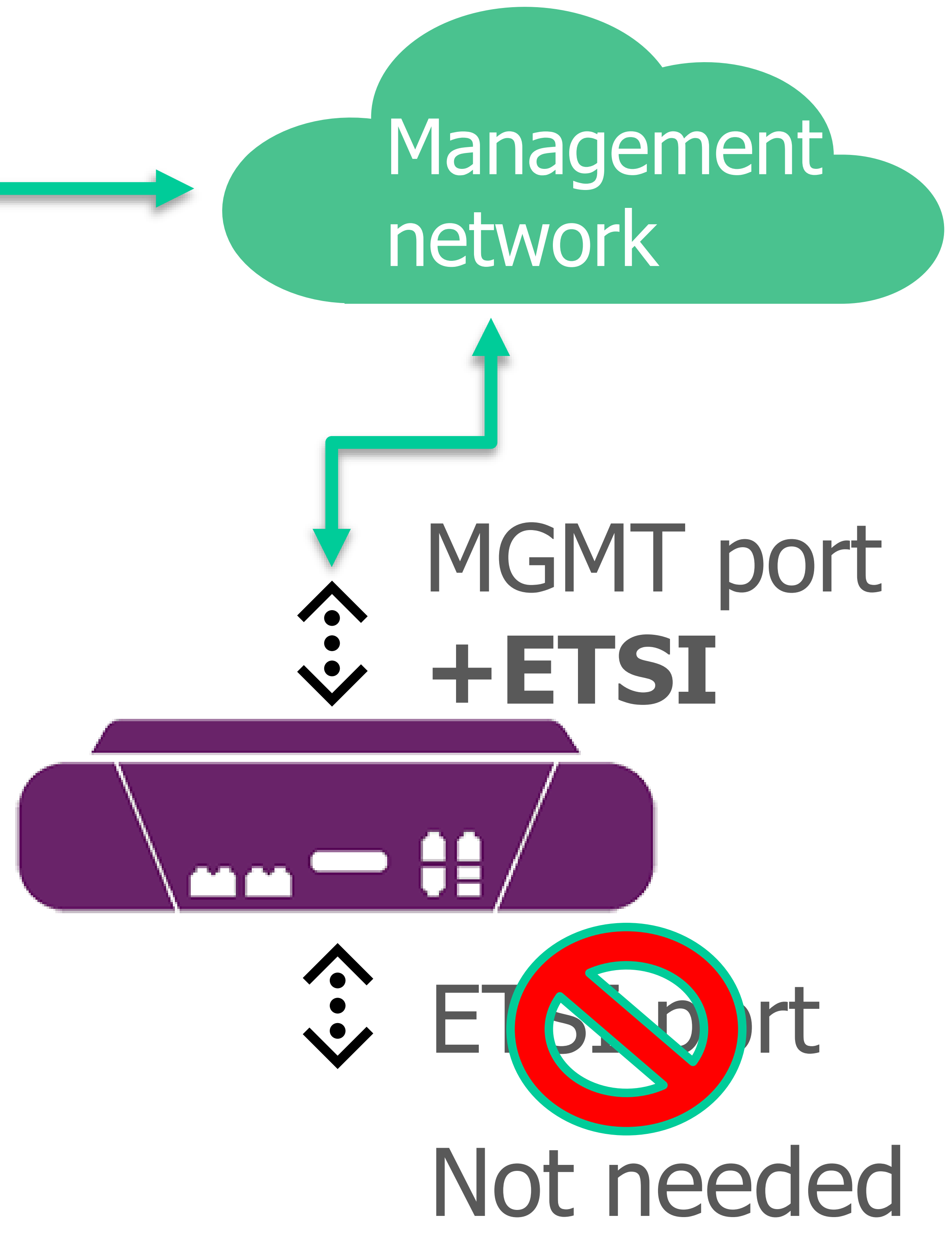
ETSI Cient



ETSI client

This topology concerns encryptors that have **only one port (ETSI+ Management)**

→ ETSI QNC port is connected to the management LAN



DO IT FOR EACH QNC

ETSI client

SWISS QUANTUM

Files requirement:
"Best practices" - RECOMMENDED

- Customer **must** own the files related to their "client side"
  - ❑ CA cert file Generated by customer, same for both encryptors
  - ❑ .key file (private key) One per client encryptor
  - ❑ .cert file (public key) One per client encryptor

- Customer **should** own the files related to the QKD device
  - ❑ CA cert file Generated by customer, same for both QNCs
  - ❑ .key file (private key) One per QNC
  - ❑ .cert file (public key) One per QNC

Recommend 1 CA file covering both Client and QKD certs/keys

ETSI client

Files requirement:
Alternative "Optional for **test** only"

- IDQ provides 9 files:
  - ❑ CA cert files **Same** for **encryptors and QNCs**
  - ❑ For each encryptor:
    - ❑ .key file (private key)
    - ❑ .cert file (public key)
  - ❑ For each QNC:
    - ❑ .key file (private key)
    - ❑ .cert file (public key)

ETSI client

Understand the ETSI TLS process

Client

Customer **Certification Authority**
Customer CA file (cert file)
ie: ClientCA.cert.pem

IDQ

IDQ **Certification Authority**
IDQ CA file (cert file)
ie: IDQCA.cert.pem

Each of the parties must have and operate <u>his own CA</u>

They are <u>independent</u>, nobody knows each other

ETSI client

Understand the ETSI TLS process

Client

Customer Certification Authority
Customer CA file (cert file)
ie: ClientCA.cert.pem

Based on its CA, the customer generates 2 files: cert file and key file.

ie: ClientETSI.key.pem / ClientETSI.cert.pem

These files will be imported into the customer's encryptor

IDQ

IDQ Certification Authority
IDQ CA file (cert file)
IDQCA.cert.pem

Based on its CA, IDQ generates each QNC with 2 files: cert file and key file.

ie: kms-master.key.pem / kms-master.cert.pem

These files will be already hosted into the QNC

Understand the ETSI TLS process

Client

own:
ClientCA.cert.pem
ClientETSI.key.pem
ClientETSI.cert.pem

1- Client gives to the QNC the client CA cert files

IDQ

own:
IDQCA.cert.pem
kms-master.key.pem
kms-master.cert.pem

Encryptor

ClientETSI.key.pem
ClientETSI.cert.pem

QNC

kms-master.key.pem
kms-master.cert.pem

ClientCA.cert.pem

ETSI client

Understand the ETSI TLS process

Client

own:
ClientCA.cert.pem
ClientETSI.key.pem
ClientETSI.cert.pem

2- IDQ gives to the client encryptor the CA cert files

IDQ

own:
IDQCA.cert.pem
kms-master.key.pem
kms-master.cert.pem

Encryptor

ClientETSI.key.pem
ClientETSI.cert.pem

IDQCA.cert.pem

QNC

kms-master.key.pem
kms-master.cert.pem

ClientCA.cert.pem

ETSI client

SWISS QUANTUM

Understand the ETSI TLS process

Client

own:
ClientCA.cert.pem
ClientETSI.key.pem
ClientETSI.cert.pem

IDQ

own:
IDQCA.cert.pem
kms-master.key.pem
kms-master.cert.pem

Encryptor

ClientETSI.key.pem
ClientETSI.cert.pem

IDQCA.cert.pem

1. Req, ClientETSI.cert.pem

Accepted

QNC

kms-master.key.pem
kms-master.cert.pem

ClientCA.cert.pem

clientETSI DN is **OK**

QNC must approve this conection request
QNC must:

- store the client's root CA
- know the DN of the client request

IDQ

SWISS QUANTUM

Understand the ETSI TLS process

Client

own:
ClientCA.cert.pem
ClientETSI.key.pem
ClientETSI.cert.pem

IDQ

own:
IDQCA.cert.pem
kms-master.key.pem
kms-master.cert.pem

Encryptor already store the root CA of IDQ

Encryptor

ClientETSI.key.pem
ClientETSI.cert.pem

IDQCA.cert.pem

1. Req, ClientETSI.cert.pem

2. Rep, kms-master.cert.pem

Accepted

QNC

kms-master.key.pem
kms-master.cert.pem

ClientCA.cert.pem

clientETSI DN is **OK**

IDQ

A simple shell script to verify ETSI connectivity

SWISS QUANTUM

```
#!/bin/sh

CURRENTDIR=$(dirname $0)

KMSM_IP='192.168.10.101:443'
KMSS_IP='192.168.10.106:443'

while true
do
date
echo "Get key round"
echo "    Get New Key from master"
NewKey=$(curl -Ss --cert client1.pem  --key client1-key.pem --cacert longTermCA.pem -k https://$KMSM_IP/api/v1/keys/ETSI2/enc_keys)
KeyID=$(echo $NewKey | jq '.keys[0].key_ID' | cut -d '"' -f 2)
echo "        Found ID: $KeyID"
echo "    Get Key with ID from slave"
Rep=$(curl -Ss --cert client1.pem --key client1-key.pem  --cacert longTermCA.pem -X POST -H 'Content-Type:application/json' -d "{\"key_IDs\":[{\"key_ID\":\"$KeyID\"}]}" -k
https://$KMSS_IP/api/v1/keys/ETSI1/dec_keys)
Key=$(echo $Rep | jq '.keys[0].key' | cut -d '"' -f 2)
KeyIDSlave=$(echo $Rep | jq '.keys[0].key_ID' | cut -d '"' -f 2)
echo "        Found Key: $Key"
echo "        Found ID: $KeyIDSlave"
echo "    Waiting 5 seconds for new request"
sleep 5
done
```

Modify Key Interface Address, SAE Address, Cert and Key Locations

Output from shell script

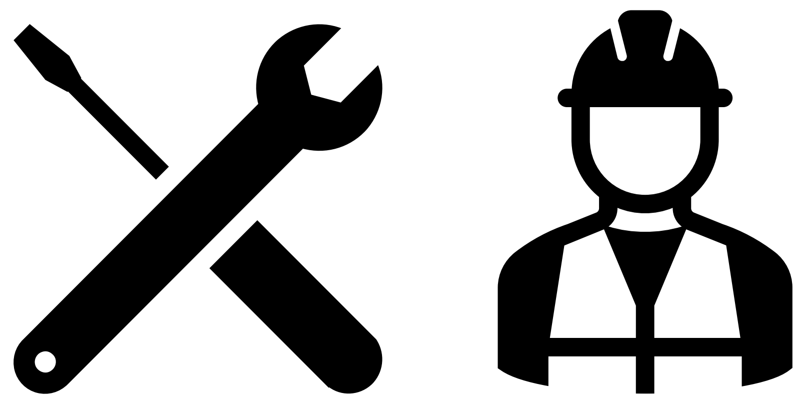
```
Tue 09 Feb 2021 01:32:40 PM EST
Get key round
    Get New Key from master
        Found ID: ebf96e46-39b6-43c9-af1b-060d7a72bf2c
    Get Key with ID from slave
        Found Key: mCJjc0Dask1MyxXMdf+UGDEQZxG7i6oMtD6g0JSo1nA=
        Found ID: ebf96e46-39b6-43c9-af1b-060d7a72bf2c
    Waiting 5 seconds for new request
Tue 09 Feb 2021 01:32:45 PM EST
Get key round
    Get New Key from master
        Found ID: 7e3ef78b-c14d-4bf7-a71a-8a2bbe36a305
    Get Key with ID from slave
        Found Key: 7PpmYQmnqeyJfi1C7D+2LImAmGXCPVL0W2icpamas8I=
        Found ID: 7e3ef78b-c14d-4bf7-a71a-8a2bbe36a305
    Waiting 5 seconds for new request
```

Verified/Tested using Ubuntu20

Key ID

Quantum Key

Troubleshooting



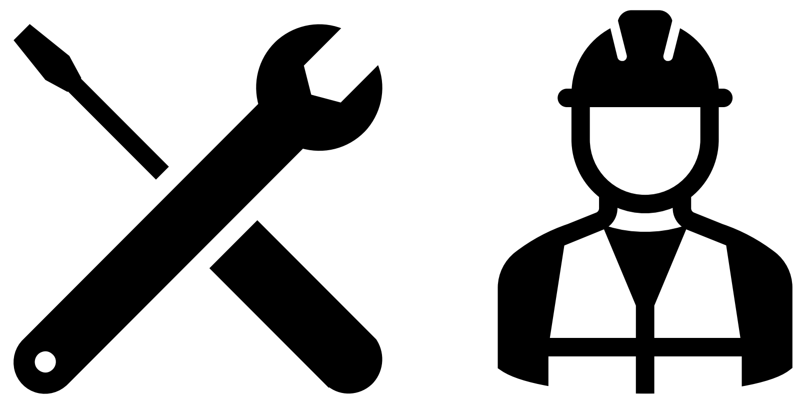
- The Cerberis XG service channel is not synchronizing (Service LED blinking):
    1. De-connect the service channel fiber (with fixed attenuator if installed)
    2. De-connect SFP module
    3. Check the SFP insertion (in both terminals)
    4. Clean the fixed attenuator connectors
    5. Connect the fixed attenuator on the SFP, after the LC/UPC fiber is inserted a clear “click” must be heard.
    6. Clean the service channel fiber connectors (in both terminals)
    7. Connect the service channel fiber to the fix attenuator, after the LC/UPC fiber is inserted a clear “click” must be heard.

Note: If the problem persists check the attenuation on the service channel fiber (SFP are designed to work for 100km fiber distance, add fix attenuator to have between 12 to 20dB attenuation on the SC)
If during the key exchange a service channel ERROR will be shown in the logs check all the points above.

- Poor correlation during the “time bin alignment”
    1. Clean the QC fiber (in both terminals)
    2. Check QC fiber connections (in both terminals), after the SC/UPC fiber is inserted a clear “click” must be heard.
    3. If the pattern looks completely “flat” (quite constant number of counts per bin) check the integrity of the QC fiber.

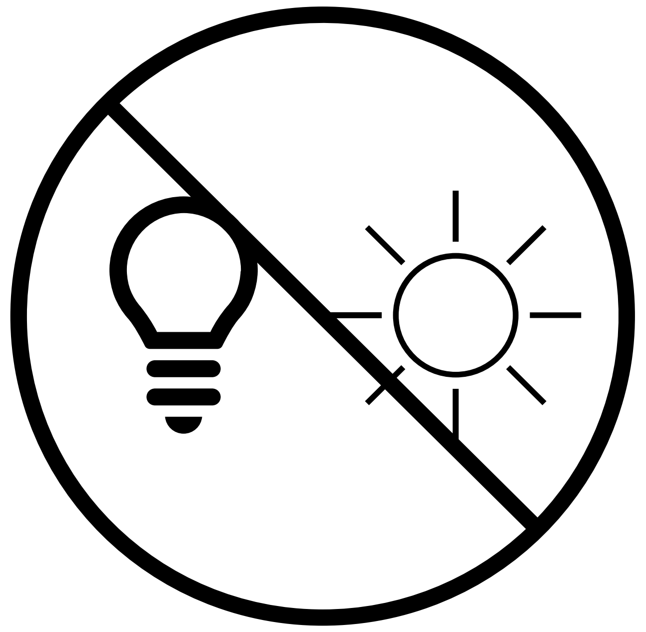
- System can arrive to “key exchanging” state, but QBER and Visibility are poor (QBER ~5%, Visibility ~90%)
    1. Check the same points as above.

Troubleshooting Cont'd



- Detection count is high during alignment or system unable to align quantum channel

1. Verify that excess light is not entering the fiber or fiber connector. Ambient light (especially sunlight) can penetrate most fiber patch cord shielding. Try turning off the lights, covering the fibers and connectors, or use armored fibers.



Power Off

To power off the QKD system, you have 2 choices:

Login to the CerberisXG (or Cerberis3) and enter the command "shutdown". Once the system halts you can remove the power.

On CerberisXG: Press the power button once. The fans will spin to maximum while the system shuts down. When the fans stop it is safe to remove the power. If the fans do not stop within 90 seconds, hold the power button down until the system halts.

On Cerberis3: Log into the Shelf Manager. Enter the command: "chassis_control -d" to shut down the chasis. The Blue LEDs will start to blink. Once the LEDs are solid blue it is safe to remove power.

Factory Default

To reset configuration to the factory settings, access the Cerberis XG vis ssh and use the command below in the QNET shell and confirm:

```
ssh admin@10.10.10.53 (same thing for 10.10.10.54)
admin> reset factory
Do you really want to reset the system to factory settings? [Y/n]Y
```

The following tasks are executed:

- Restore default KMS configuration,
  - Name: QNC
  - Database: empty
- Restore default QKD configuration files,
- Restore default QNC WebApi configuration files,
- Restore default QNET shell configuration files,
- Restore default network interface settings,
- Restore default users & passwords, groups,
- Restore default certificates and Cas,
- Cleanup KMS logs,
- Clean QKD logs,
- Clean QKD backups,
- Clean QNC WebApi logs,
- Clean QNET shell logs,
- Clean known hosts files,
- Clean temporary files,
- Clean less history.

Additional Commands

This training guide does not represent all the configuration/menu options on the CerberisXG. Please see the help menu on the QKD system for additional list of commands:

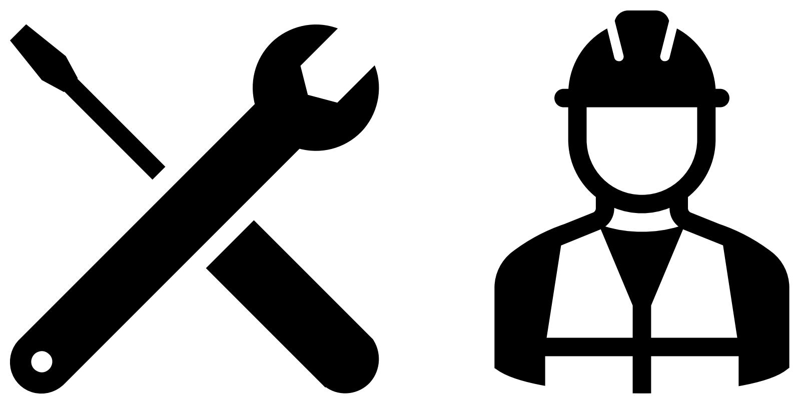
From the CLI, type the command "help".

Maintenance

SWISS QUANTUM

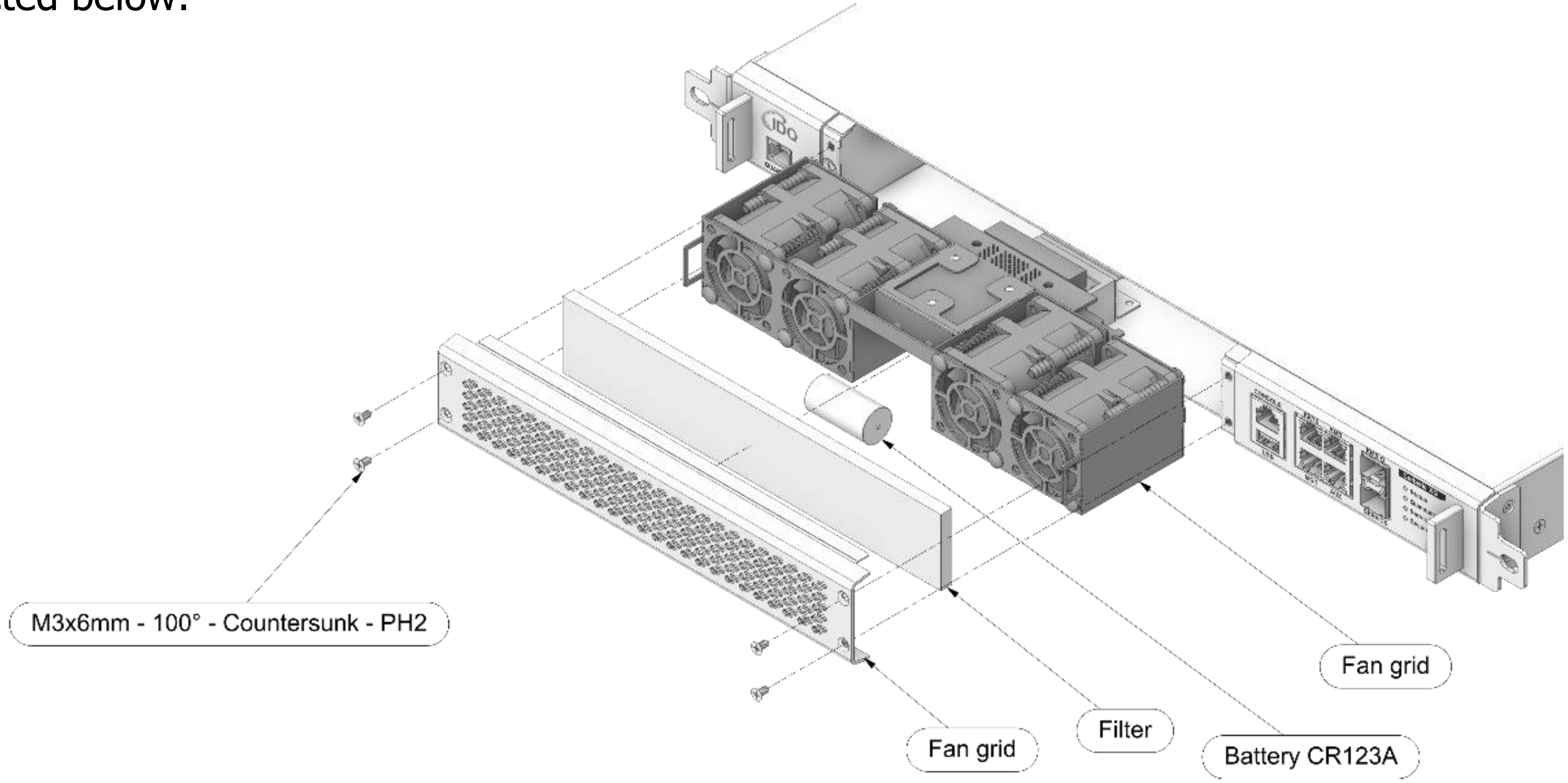
On your Cerberis XG, the following parts can be serviced in case of defect or replacement need:

- Filter (30 PPI, UL 94 HF-1, 10mm)
- Fan module (to order from IDQ)
- Battery for real-time clock (CR123A type)
- Power supply unit (PSU) (to order from IDQ)



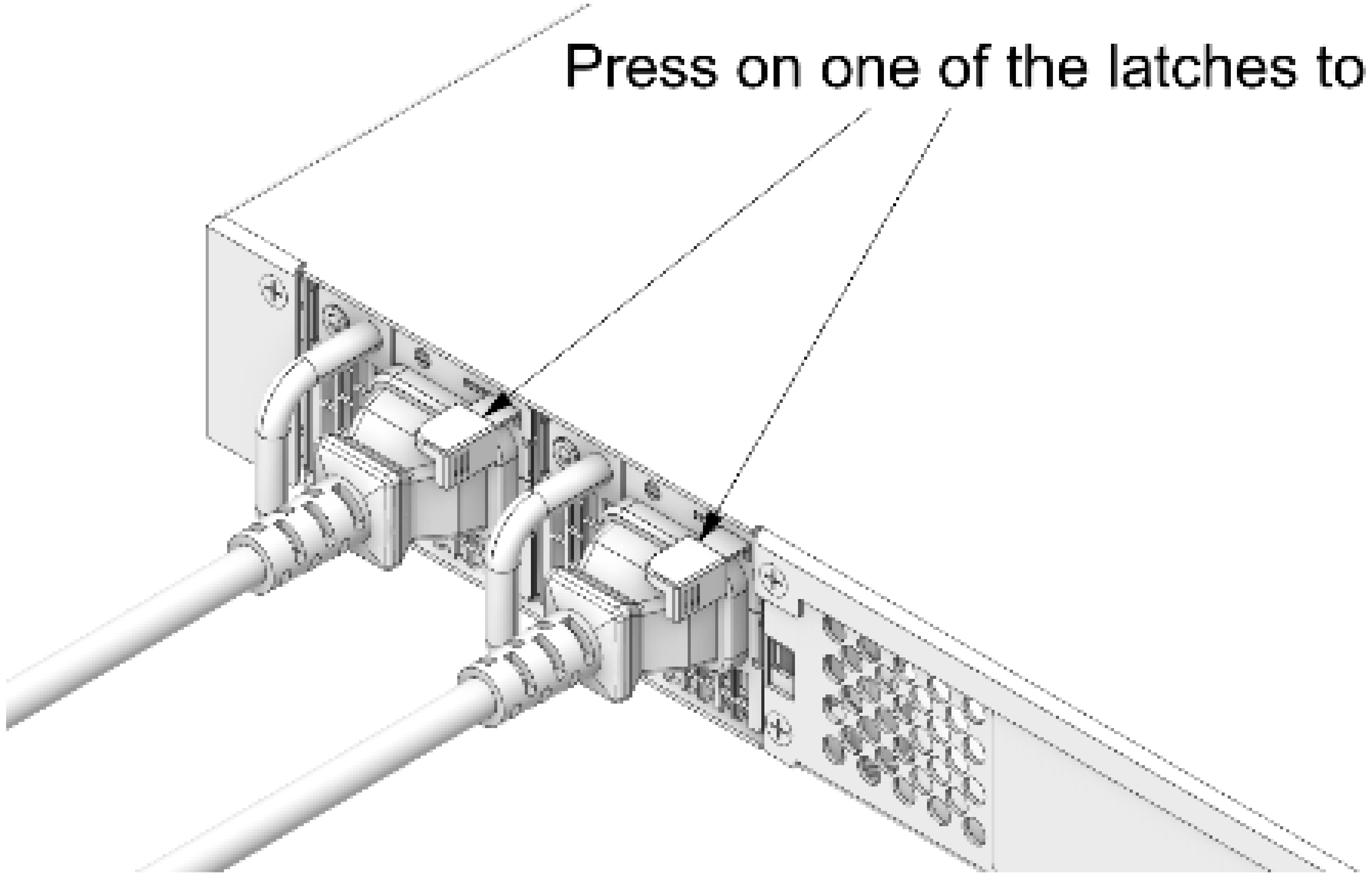
SWISS QUANTUM

The filter, the battery, and the fan module can be accessed and replaced by unscrewing the fan grid as depicted below:



Maintenance

A defective power supply unit can be hot swapped from the rear panel as instructed below:



Upgrade Migration

For customers upgrading from releases prior to 2.0, you will need to consider the following migration steps:

Follow the software upgrade procedure recommended by an IDQ Representative. Upgrades > 2.0 require a system running V1.9 or higher.

To migrate your configuration from KEMS to QMS, perform the following steps:

- Backup your QKD base configuration using the backup commands on the CLI to USB stick.
- Using the QNET tool, export your KEMS configuration using the “qnet export” command.
- Install the updated QMS and QNET per the instructions in this document.
- Using the QNET tool, import your old KEMS configuration into QMS.
- Alternatively rebuild your KMS configuration from scratch using the QNET scripts discussed earlier.

Research-Specific Functions

Research-Specific Functions – Overview of steps

Cerberis XGR/Clavis3

- Get/Set QKD Parameters

  1. On QNET activate Research User Group :

```
[idq@walsingham ~]$ qnet list role
  ID     NAME        DESCRIPTION                     ENABLED
  0002   Crypto                                      Yes
  0003   Monitor                                     Yes
  0004   Research    Role used for Research users    No
  1      Admin                                       Yes
  2      User                                        Yes
  3      SuperAdmin                                  Yes
[idq@walsingham ~]$ qnet update role Research --enabled
KMS role Research successfully updated.
  ID     NAME       DESCRIPTION                    ENABLED
  0004   Research   Role used for Research users   Yes
[idq@walsingham ~]$
```

  2. On QNET add a new User into Research User Group :

```
[idq@walsingham ~]$ qnet create user seddick -r Research
Password: *****
Password doesn't match password policy.
[idq@walsingham ~]$ qnet create user seddick -r Research
Password: ***************
KMS user seddick successfully created.
  ID      USERNAME   EMAIL   ROLES      ENABLED
  U1005   seddick            Research   Yes
[idq@walsingham ~]$
```

1. On QNET Deploy Node Config :

```
[idq@walsingham ~]$ qnet deploy node KMSXGM1
deploying KMSXGM1: .....................
KMS node KMSXGM1 (10000001) successfully deployed. (group TestGrp)
```

1. On QNET use QKD show Parameters to display the QKD parameters:

4. qnet show qkd XXX -url https://localhost:8443
5. Authentication: Certificate qnetprod
6. PARAMETER VALUE
7. /CommunicatorAlice/FPGA/Distillation@CompressionRatio 30
8. /CommunicatorAlice/FPGA/Model@Filter true
9. /CommunicatorAlice/FPGA/RNG@QRNG true
10. /CommunicatorAlice/Optics/Laser/TargetPower@PhotonNumber 0.03
11. /CommunicatorAlice/Optics/Pulse@Width 500
12. /CommunicatorAlice/Bists@Enabled true
13. /CommunicatorAlice/Alignment@PhotonNumber 0.08
14. /CommunicatorAlice/Regulation/DATA@Darkcounts 0
15. /CommunicatorAlice/Regulation/DATA@Deadtime 1
16. /CommunicatorAlice/Regulation/MONITOR@Darkcounts 0
17. /CommunicatorAlice/Regulation/MONITOR@Deadtime 1
18. /CommunicatorAlice/Regulation/QBER@IntegrationTime 10
19. /CommunicatorAlice/Regulation/Visibility@IntegrationTime 10
20. QKD provider XXX successfully returned

Authenticate as SuperAdmin First!

On the linux prompt, first authenticate to QNET/QMS using the SuperAdmin user as follows:
qnet config qnetwebapi -url https://localhost:443 -user SuperAdmin -pwd SuperAdmin

Enable the Research functionality:
qnet update role Research --enabled

Create a new Research user:
qnet create user testuser -r Research

```
[asauzeat@ross qnet-package]$ qnet config configwebapi -url https://localhost -user SuperAdmin -pwd SuperAdmin
QNET WebApi: https://localhost (Available)
Version: 1.1.133
Authentication: Basic
User: SuperAdmin
[asauzeat@ross qnet-package]$ qnet update role Research --enabled
KMS role Research successfully updated.
  ID    NAME      DESCRIPTION                 ENABLED
  0004  Research  Role used for Research users  Yes
[asauzeat@ross qnet-package]$ qnet list role
  ID    NAME        DESCRIPTION                 ENABLED
  0002  Crypto                                    Yes
  0003  Monitor                                   Yes
  0004  Research    Role used for Research users  Yes
  1     Admin                                     Yes
  2     User                                      Yes
  3     SuperAdmin                                Yes
```

Re-Authenticate as the Research User

On the linux prompt, authenticate to QNET/QMS using the new research user previously created as follows:
qnet config qnetwebapi -url https://localhost:443 -user testuser -pwd A45kK!@Lkd99%^123

Use “qnet list role” to verify active users and corresponding roles.

Update Parameters

Cerberis XGR

- Get QKD Parameters
  1. On QNET use **QKD show QKD** to show the QKD parameters:
     - qnet show qkd <QKD_NAME> -url <QKD_IP:8443>

       example :qnet show qkd QKDA –url https://192.168.10.102:8443

       or

       qnet show qkd QKDA

- Set QKD Parameters
  1. On QNET use **QKD update QKD** to edit the QKD parameters:
     - qnet update qkd <QKD_NAME> <PARAM_NAME> <VALUE>

       example : qnet update qkd NOMQKD /CommunicatorBob/FPGA/Model@Filter false

MAKE SURE TO SHOW AND WRITE DOWN THE ORIGINAL VALUES BEFORE CHANGING THEM!!!
You will likely want to change them back. System may not function properly after you change the values!

Cerberis XGR/Clavis3 – Customizable parameters

Component			Terminal	Value 1 or min value	Value 2 or max value	units
FPGA	Distillation	CompressionRatio	Alice/Bob	5	30	
FPGA	Model	Filter	Alice/Bob	TRUE	FALSE	
FPGA	RNG	QRNG	Alice/Bob	TRUE	FALSE	
Optics	Laser	PhotonNumber	Alice	0.001	0.1	
Optics	Pulse	Width	Alice	300	600	ns
Alignment		PhotonNumber	Alice	0.001	0.1	
Regulation	DATA	Darkcounts	Alice	0	20000	
Regulation	DATA	Deadtime	Alice	1	100	µs
Regulation	MONITOR	Darkcounts	Alice	0	20000	
Regulation	MONITOR	Deadtime	Alice	1	100	µs
Regulation	QBER	IntegrationTime	Alice	2	30	s
Regulation	Visibility	IntegrationTime	Alice	2	30	s
Bists		Enabled	Alice	TRUE	FALSE	
Optics	Detectors	Deadtime	Bob	1	**100**	µs
Optics	DataPulse	Width	Bob	300	800	ns
Optics	MonitorPulse	Width	Bob	300	800	ns
Alignment		MinDetections	Bob	500	4000	
Optimization		MinDetections	Bob	500	4000	
Optimization	ModulatorBias	Step		0.025	0.1	
Regulation		IntegrationTime	Bob	2	30	s
Bists	DetectorsBist	Timeout	Bob	600	1800	s

Disclaimer: Even though the XGR offers the possibility to change these parameters the only one for which the system functioning is guaranteed are the "FACTORY" values

IDQ4P API – Raw Key Streaming
See Supplemental Documentation

Thank You