SWISS QUANTUM

IDQ

FROM VISION TO TECHNOLOGY

Quantum Key Distribution Training Cerberis 3

Version 14 July 2021

For Q1-2021 (V1.9.0 version)

ID Quantique PROPRIETARY

IDQ

Company presentation

SWISS QUANTUM

Founded in 2001

By 4 quantum physicists from the University of Geneva

Geneva, Switzerland
Seoul, South Korea
Washington-Boston, USA
Bristol, UK

~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 part

- Be able to understand the purpose of the QKD systems
- Be able to understand the different fibers used
- Be able to understand the purpose of the KEMS server

Cerberis 3 presentation

- Commercial name: **Cerberis 3**
- Type: **QKD** (Quantum Key Distribution)
- 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)

Cerberis 3 presentation

SWISS QUANTUM

Key Markets

- Telecom and Network Operators
- Financial Services Companies
- Governments and Defence
- Healthcare Organisations
- Critical Infrastructure
- IP-rich Enterprises

Key Applications

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

IDQ

Cerberis 3 presentation

Model	Cerberis[3]
KEY FEATURES	
Key generation rate	1.25 GHz pulse repetition rate
High speed hardware-based key processing, to distill the secret keys	✓
Key security parameter[1]	$\varepsilon_{QKD} = 4.\ 10^{-9}$
Maximum transmission loss	12 dB / 14 dB / 16 dB / 18 dB /
Maximum length of quantum channel (typ. @ 0.24 dB/km)	50 km / 58 km / 66 km / 75 km /
Secret key rate	1.4 kb/s (12 dB)
PHYSICAL PARAMETERS	
Dimensions[2]	19'' rackmount 6U ATCA chassis; 13'' depth
Switching Shelf Manager (SSM)	Full status LEDs, 10GbE SFP+, GbE ports, RS-232 de-bug ports, and Telco alarm
Power supply	Up to 4x swappable 1300 W AC power supplies or 2x 90 Amp DC Power Entry Modules. The input voltage is from 100 to 240 VAC or -36 to-72 VDC
Weight for one node[2]	30 kg
Operating conditions	
Temperature	10 to 30°C
Max relative humidity (@30°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	Temperature (high and low), fan failure, power supply failure, low key rate, high QBER (intruder alarm), service and key channel failure, key buffer
Continuous monitoring	Temperature, fan and power supply operations, system uptime, firmware version, CPU load, memory usage, actual quantum key rate, QBER, compression rate (due to key processing)

[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

QKD - Basics

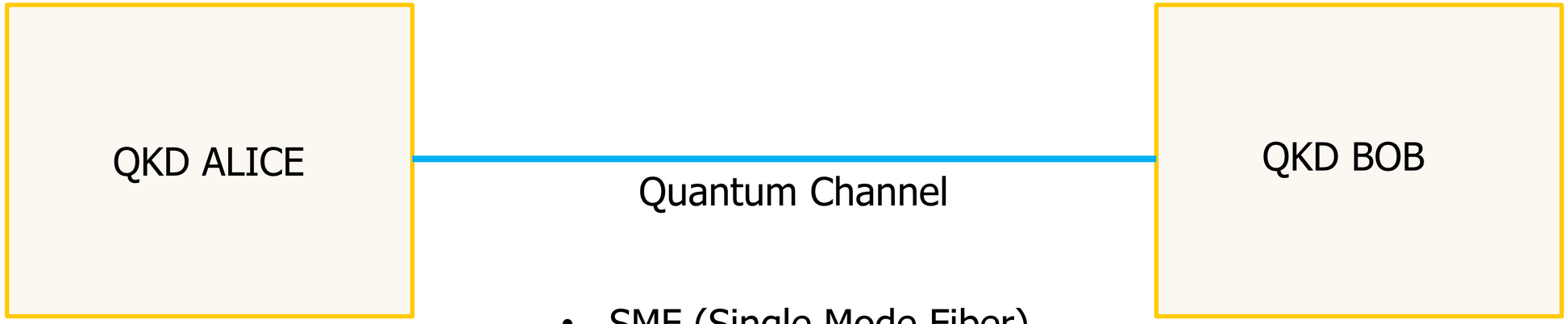
- Used **to exchange quantum bits** through a Quantum Channel
- Due to the nature of Quantum Channel, **any trial of hacking is detected**
- Quantum Channel **must use dark fiber – See IDQ Rep for alternate WDM config**
- Quantum Channel **cannot work** through **actives devices**
- Quantum Channel cannot be **used for long distances or worldwide networks**
- QKD Cerberis 3 system **doesn't encrypt data**
- QKD Cerberis 3 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 key**s to third party devices (Encryptors or HSM)
3. Third party devices **use these keys to encrypt data**

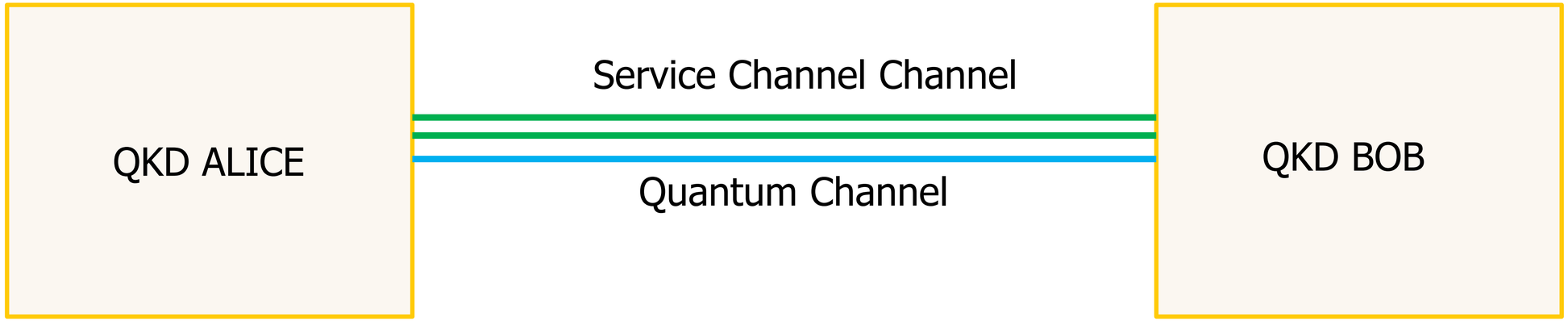
QKD - Topology



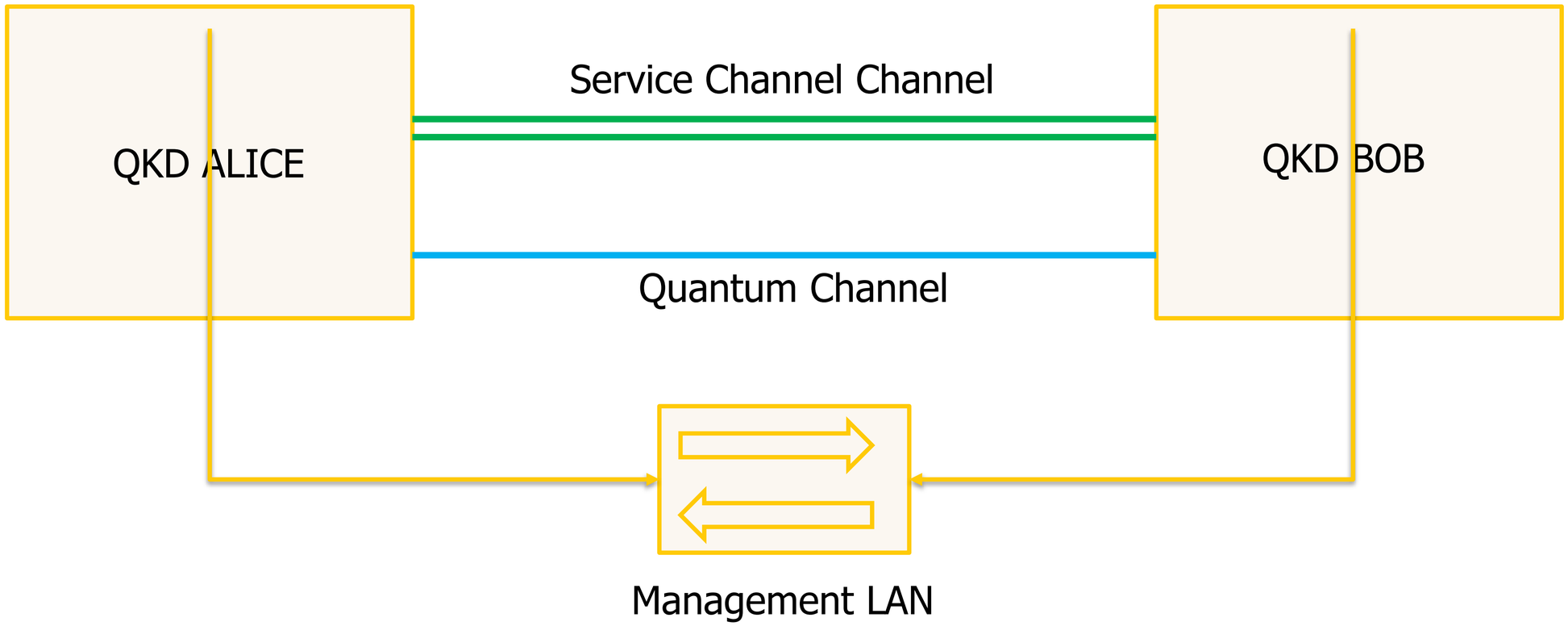
- SMF (Single Mode Fiber)
- Uses quantum physics
- Losses need to be minimized
- No active device between

QKD - Topology

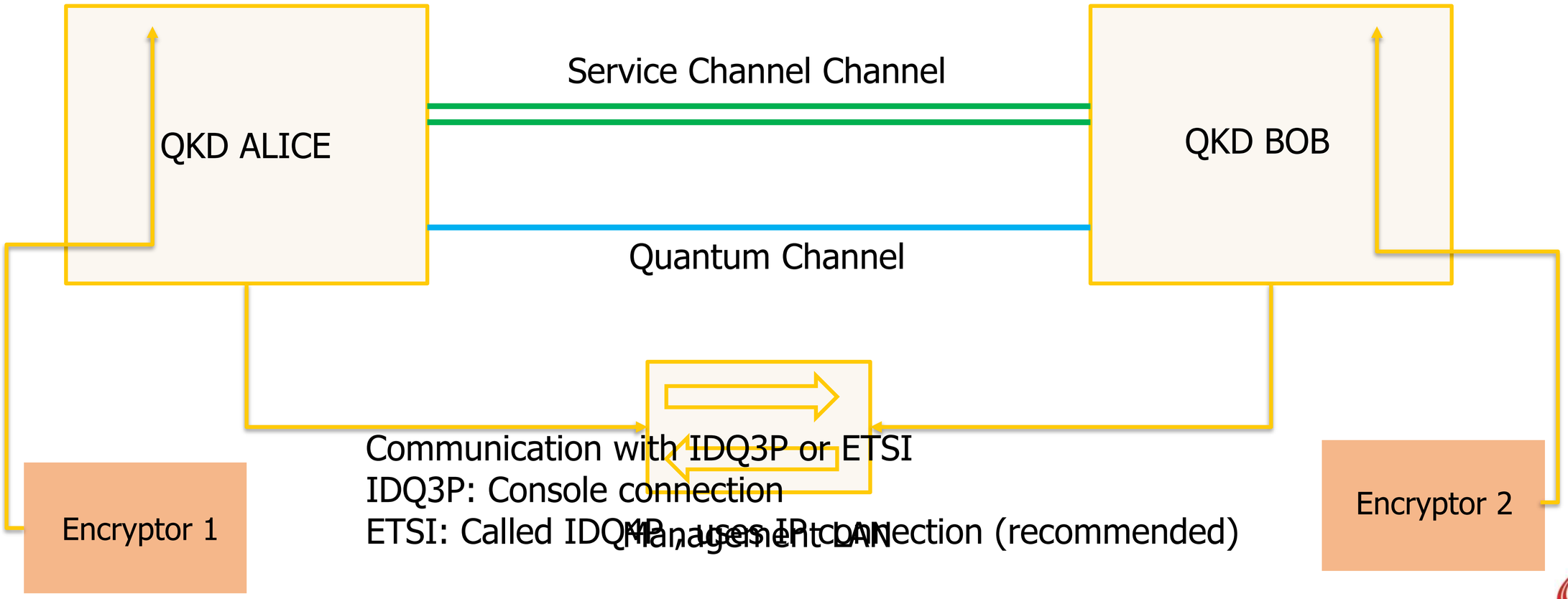
- Dual SMF (Single Mode Fiber)
- 2.67 Gbps rate
- Used for QKD clock synchronization and post-processing



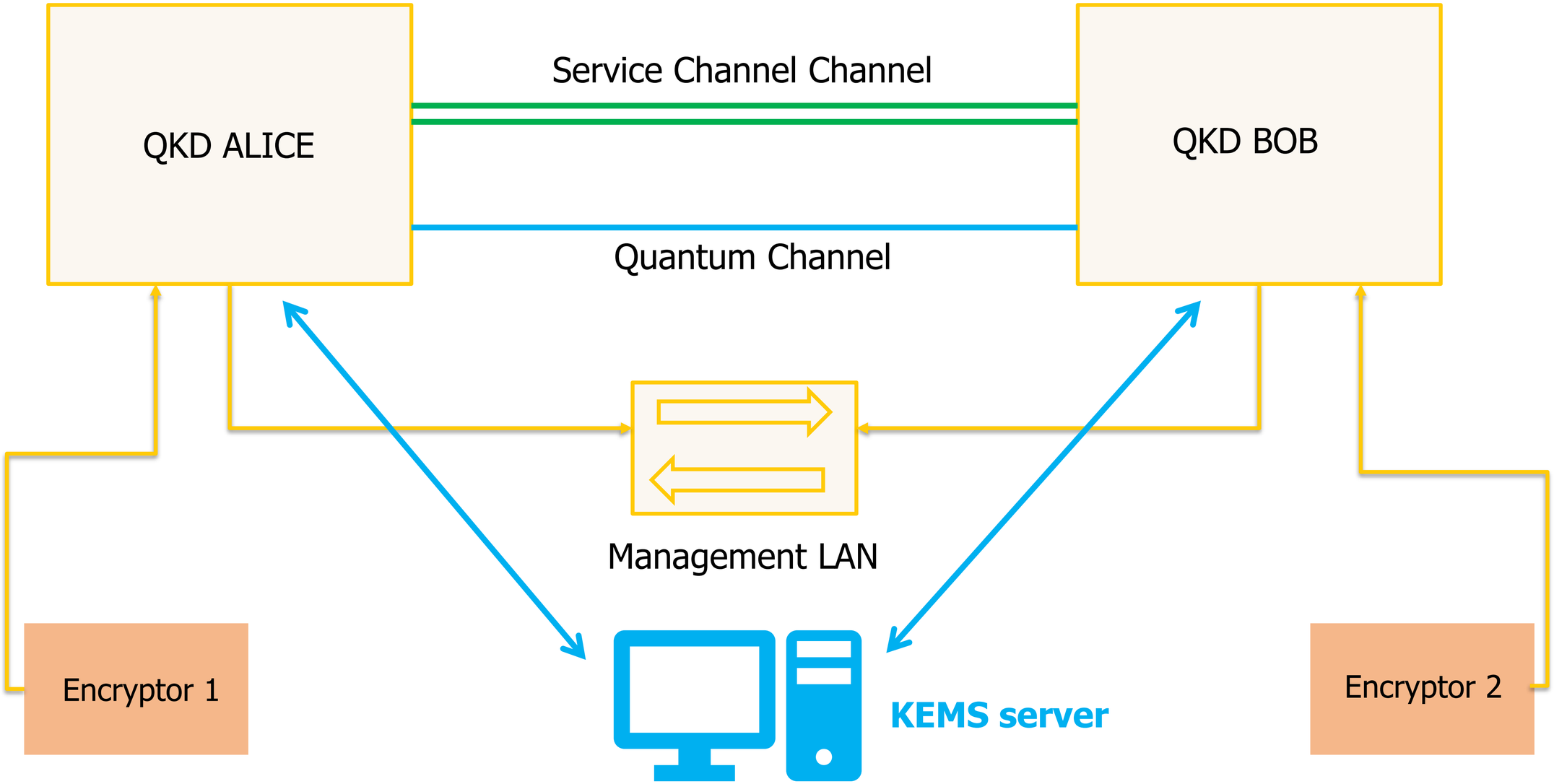
QKD - Topology



QKD - Topology



QKD - Topology



QKD - KEMS

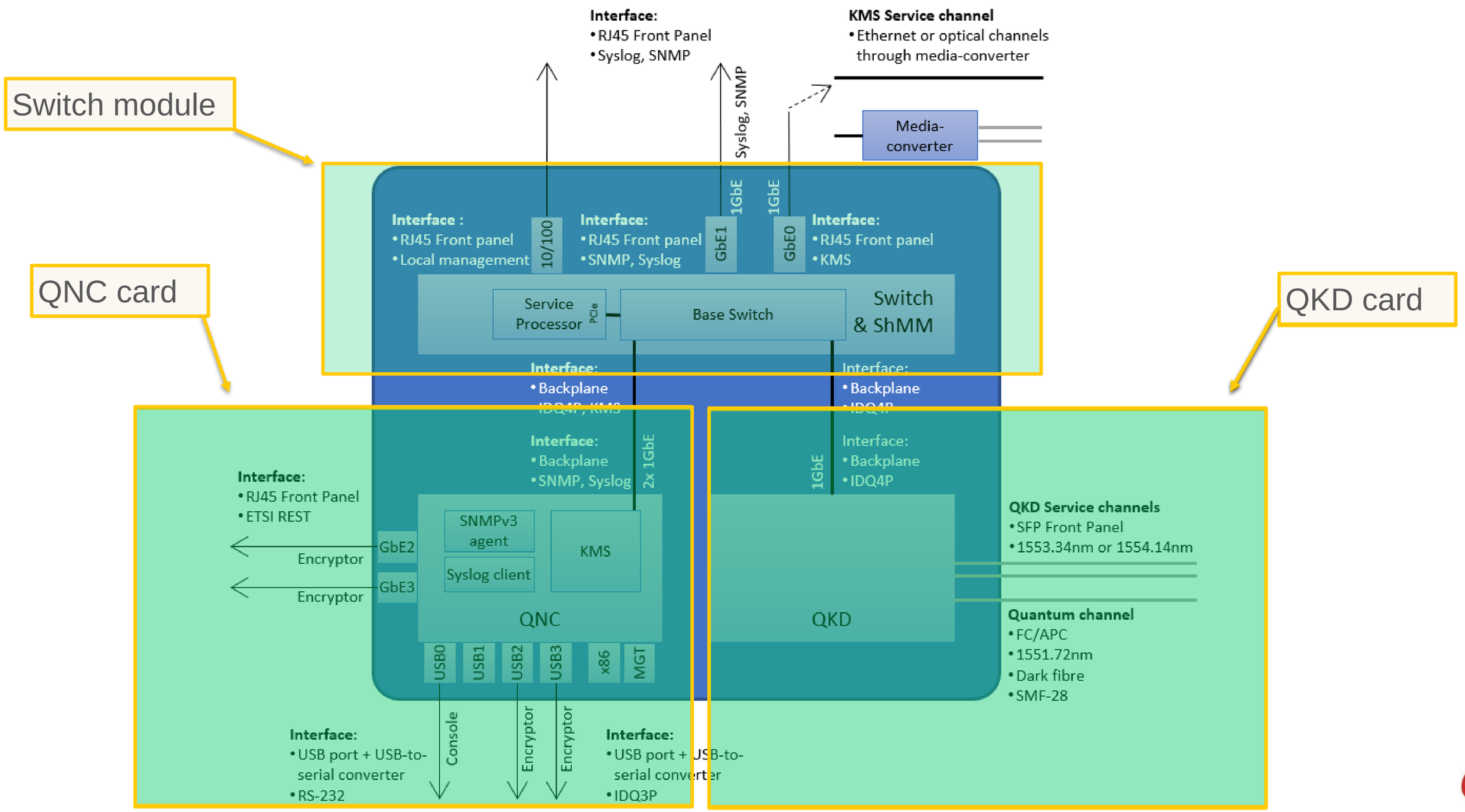
- KEMS: **K**ey **E**ntity **M**anagement **S**ystem
- **Mandatory** to make the QKD system function properly (since 2019 Q3 V1.2)
- Used to configure the QKDs, the paths and the encryptors interconnection
- Can be used on a **virtual machine** (VM) or physical server
- Low server performance requirement
- Simple deployment with Docker

QKD chassis element

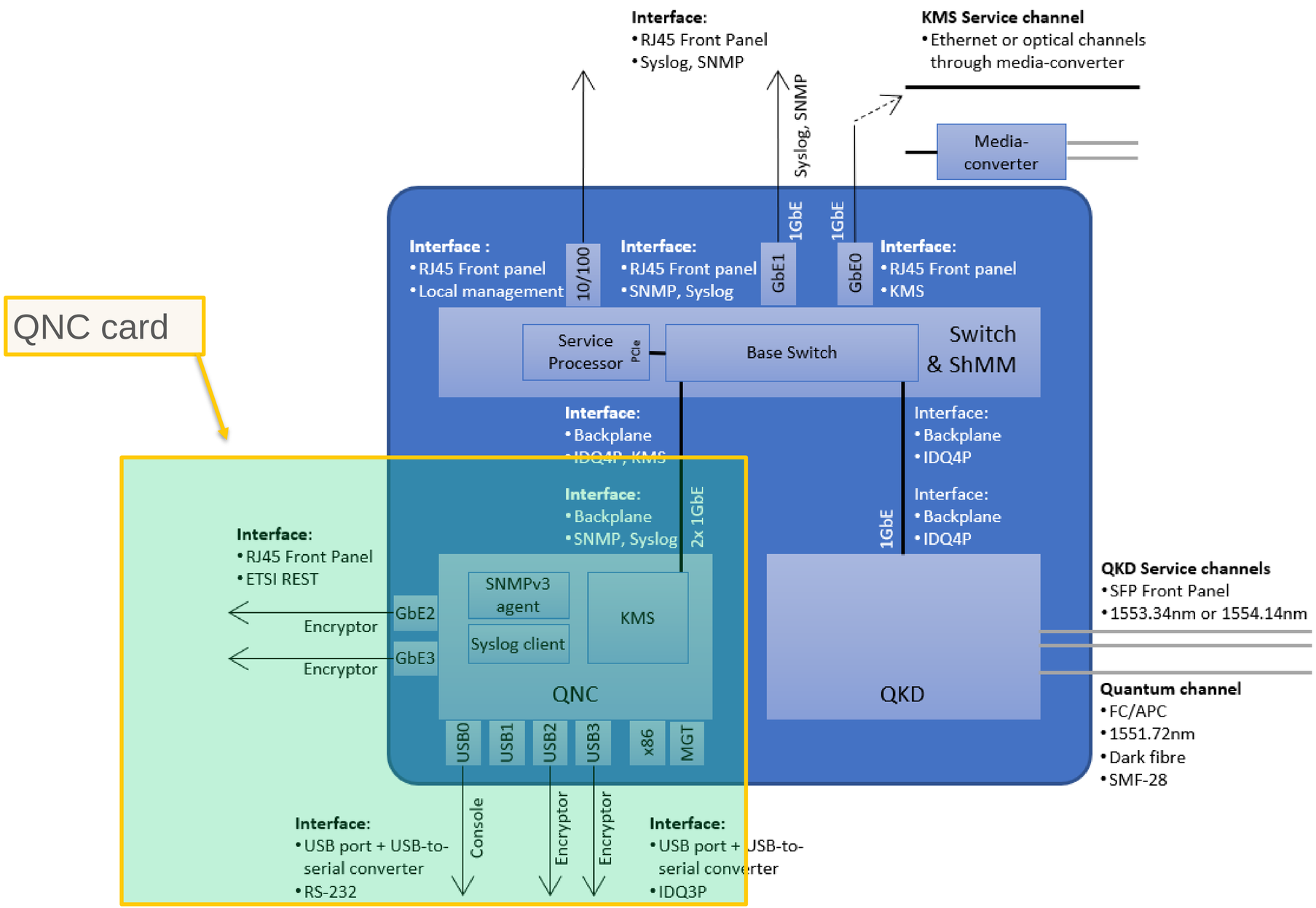
Goal of this part

- Be able to understand the purpose of the QNC card
- Be able to understand the purpose of the QKD card
- Be able to understand the purpose of the Switch module

Whole ATCA chassis diagram



The Cerberis³ system – QNC card



The Cerberis^3 system – QNC card

QNC: Quantum Node Controller

The Cerberis^3 system – QNC card

QNC card

- 1 slot module
- Linux system based
- Used for
  - Monitoring
  - Management
  - KMS User Key Agents
- QNC talks internally to QKD using the backplane

The Cerberis^3 system – QNC card

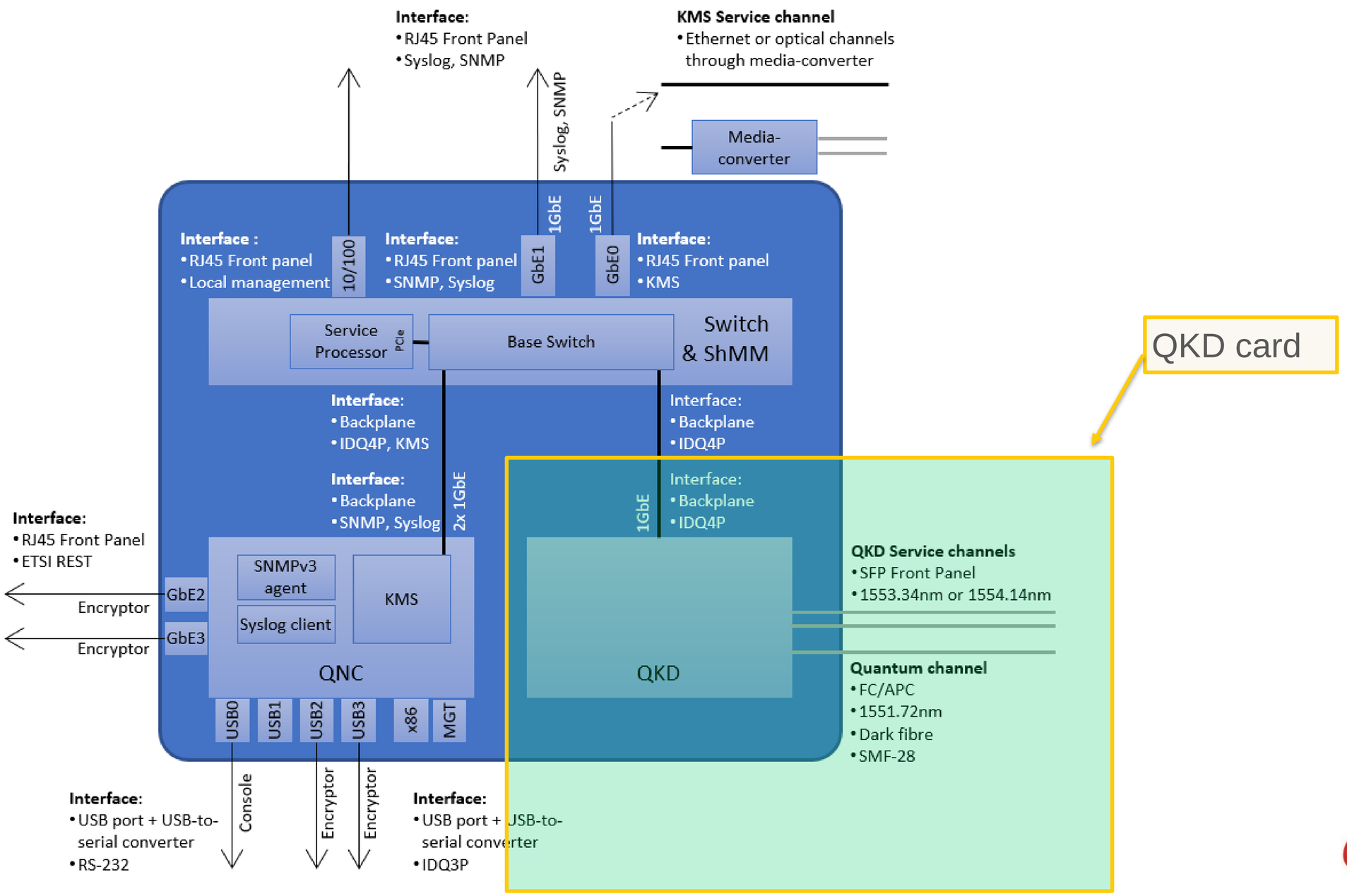
QNC ETSI
Key Interfaces

Serial ports

H/S LED

USB ports:
- Upgrade
- QNC IDQ3P key delivery interfaces

The Cerberis^3 system – QKD card



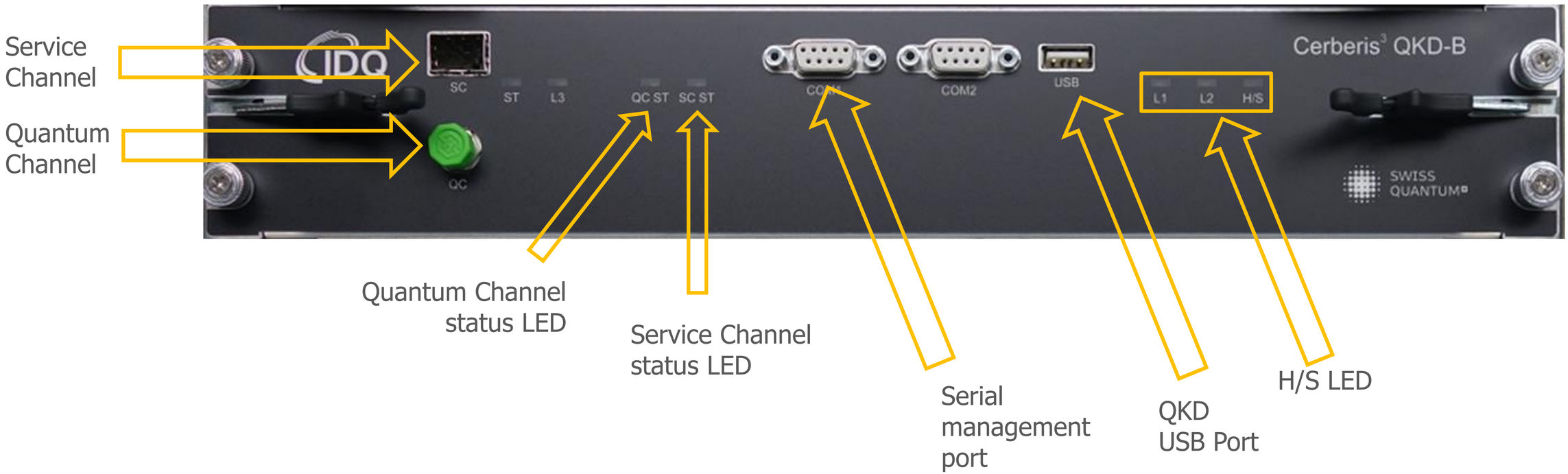
The Cerberis^3 system – QKD card

QKD: Quantum Key Distribution

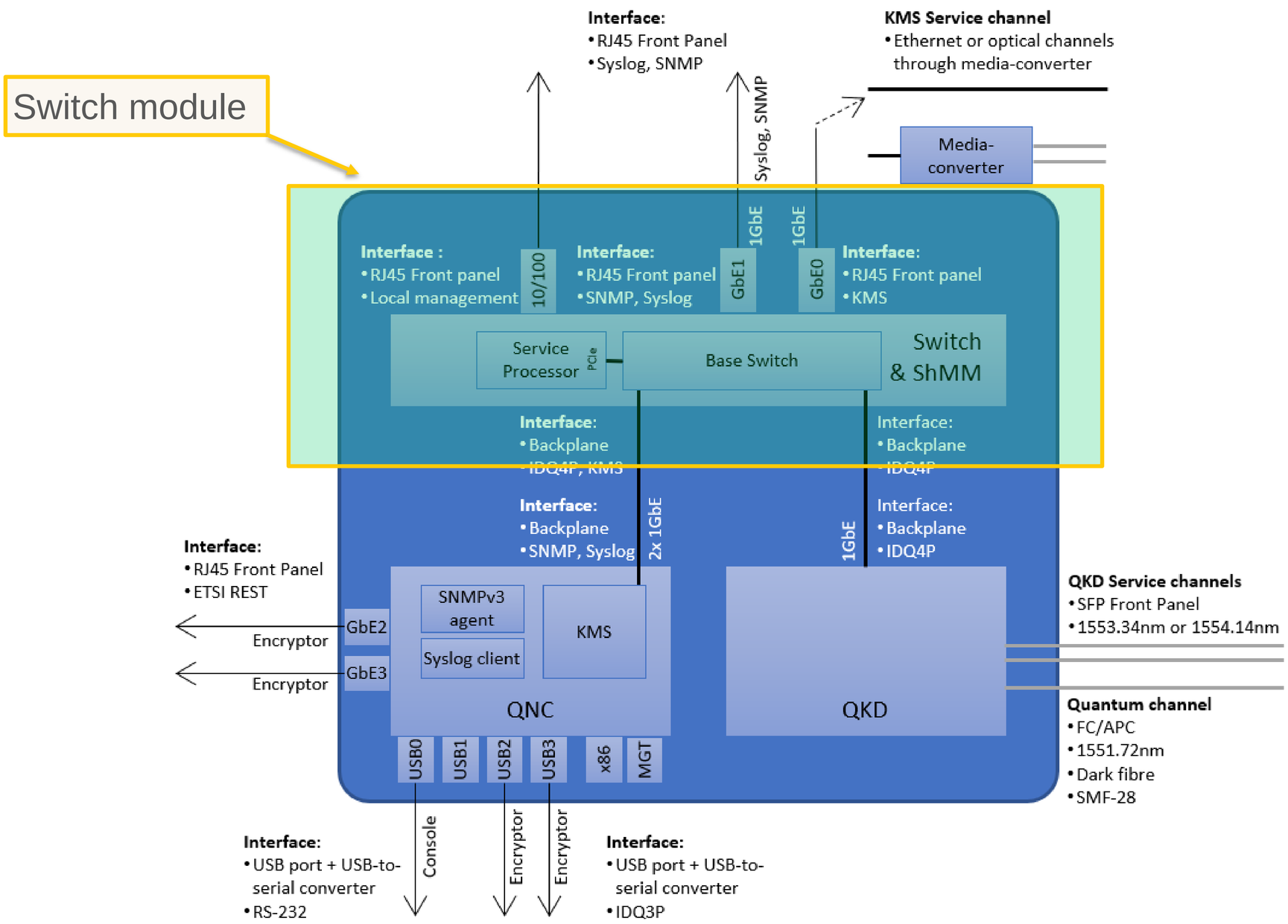
The Cerberis^3 system – QKD card

QKD card

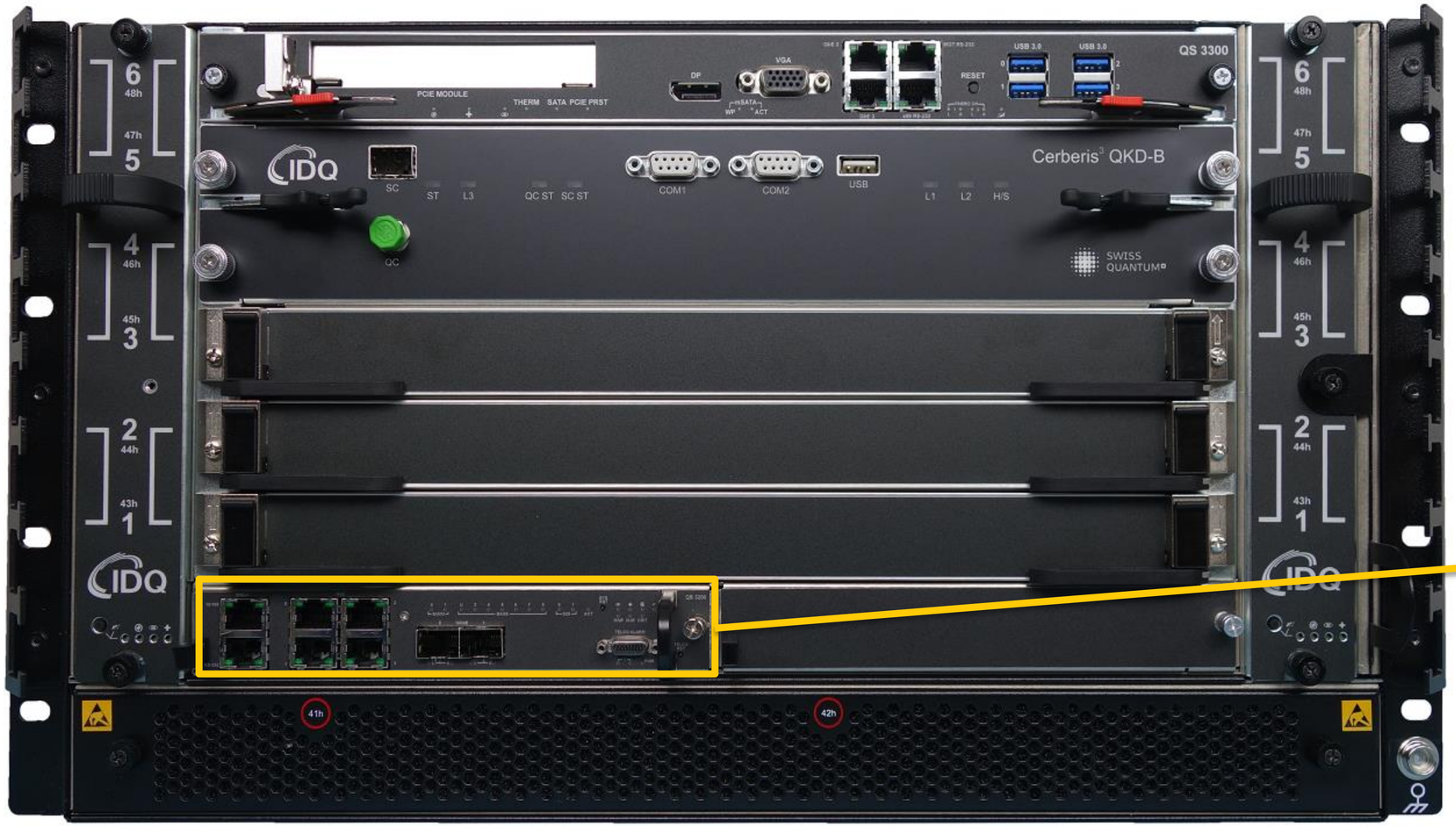
- 2 slots module
- Linux system based
- Used for Quantum Key Distribution
- It uses Quantum Physics
- QKD talks to QNC internally


Service Channel
Quantum Channel
IDQ
SC
ST
L3
QC
QC ST
SC ST
COM1
COM2
USB
L1
L2
H/S
Cerberis³ QKD-B
SWISS QUANTUM
Quantum Channel status LED
Service Channel status LED
Serial management port
QKD USB Port
H/S LED

The Cerberis³ system – Switch card



The Cerberis^3 system – Switch card



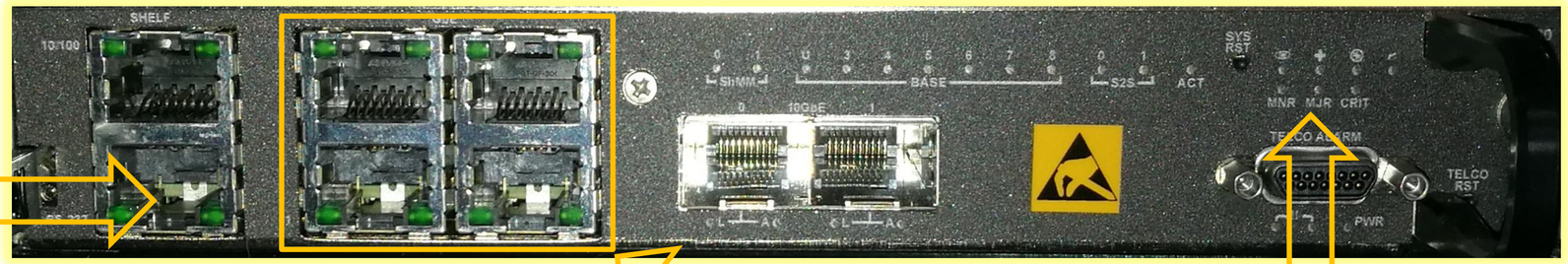
Shelf Manager

The Cerberis^3 system – Switch card

Switch card

- 1/2 slot module
- 10GbE interfaces
- Layer 2 or Layer 3 managed switch performance
- SFP+ ports and Ethernet port
- Switch used for internal communication (backplane) and the ATCA chassis management

The Cerberis³ system – Switch card



Serial management port

Switch Ethernet Ports

H/S LEDs

Cerberis 3 presentation

Fan modules

CERBERIS 3

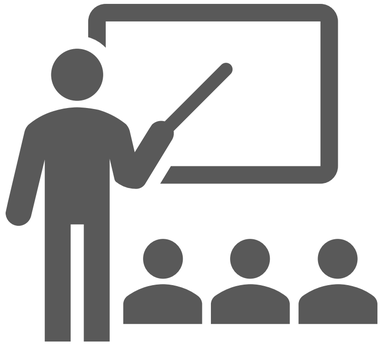
Physical Installation step by step

ID Quantique PROPRIETARY

Cerberis 3 – 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 properly insert the QNC and QKD cards
- Be able to make sure the system is properly started
- Be able to clean and connect the different fibers



Safety precautions

- Carefully read through this procedure before operating the Cerberis3 Node 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.

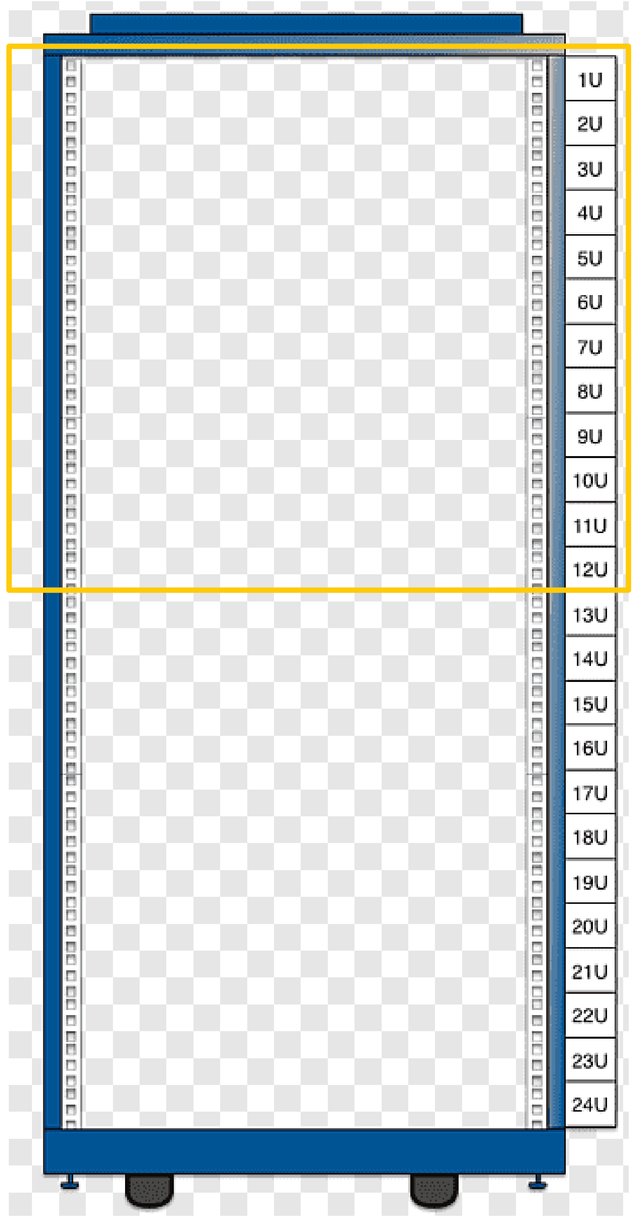
Cerberis 3 – Physical setup (Rack mount)

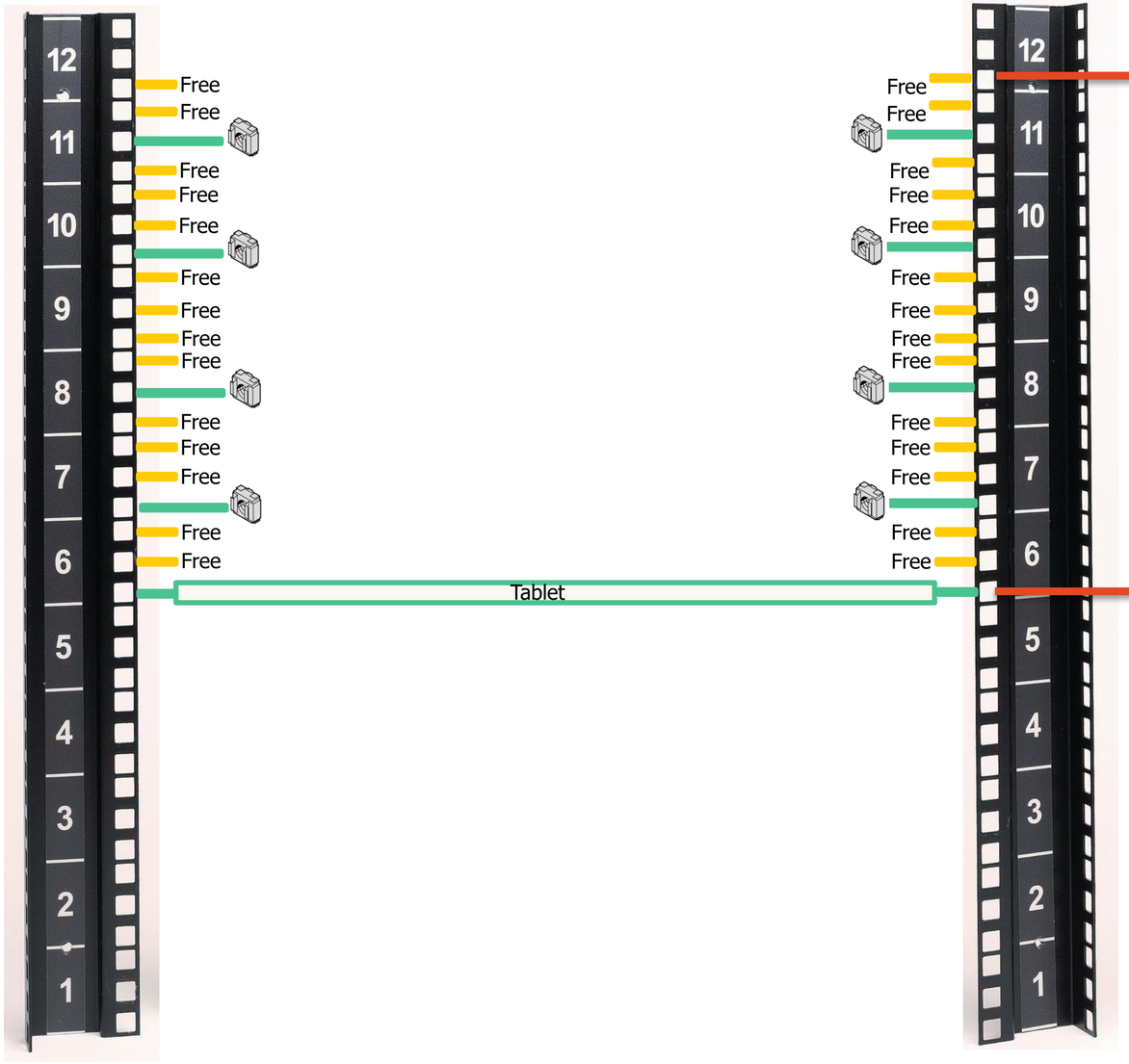
8 Screws needed

Tablet or support strongly recommended

Chassis is quite heavy

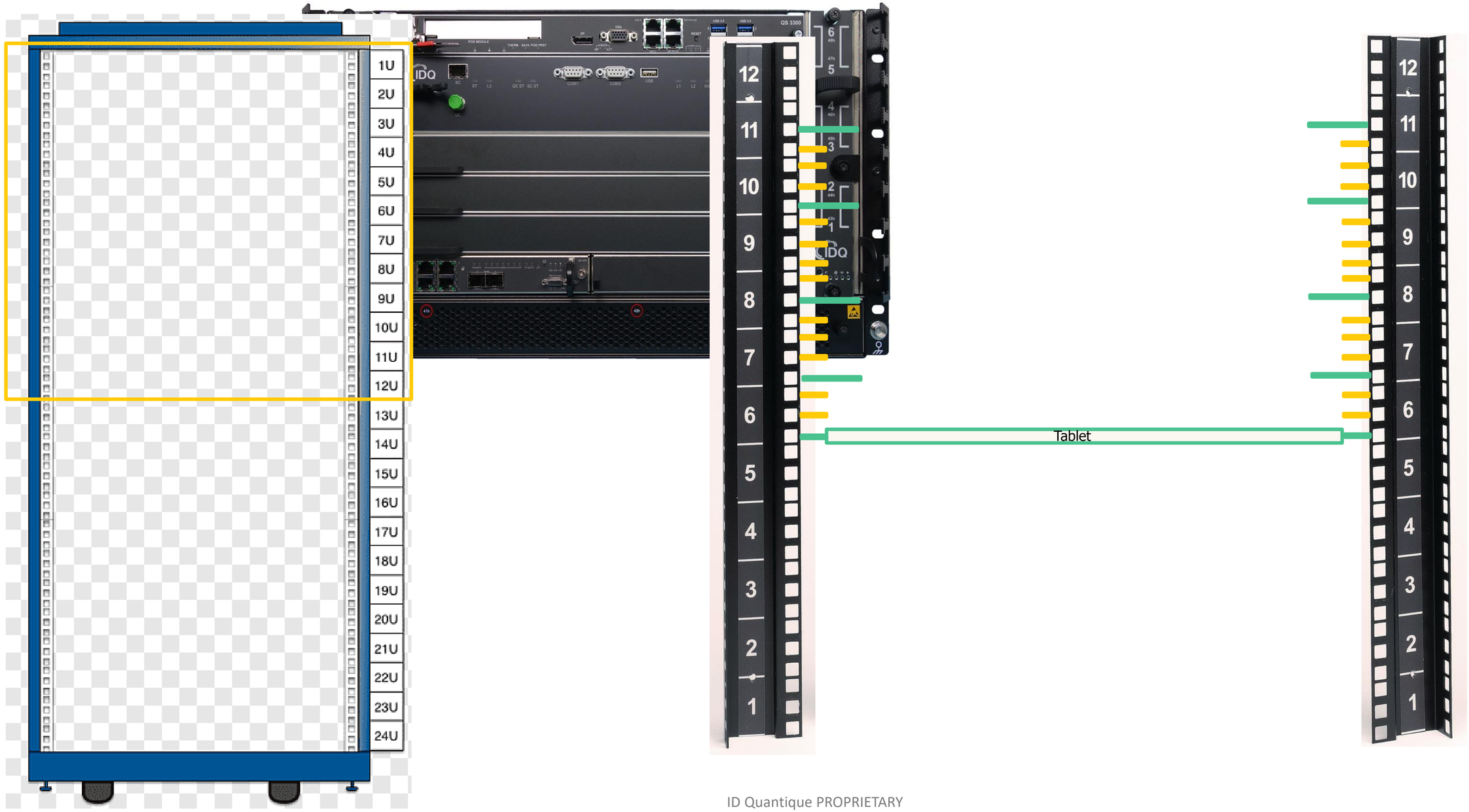
Cerberis 3 – Physical setup (Rack mount)



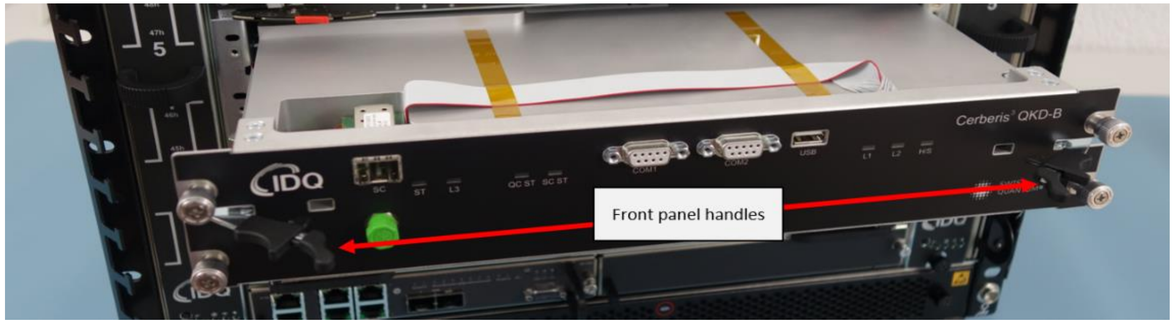


Almost 6 U needed + 1 tablet

Cerberis 3 – Physical setup (Rack mount)



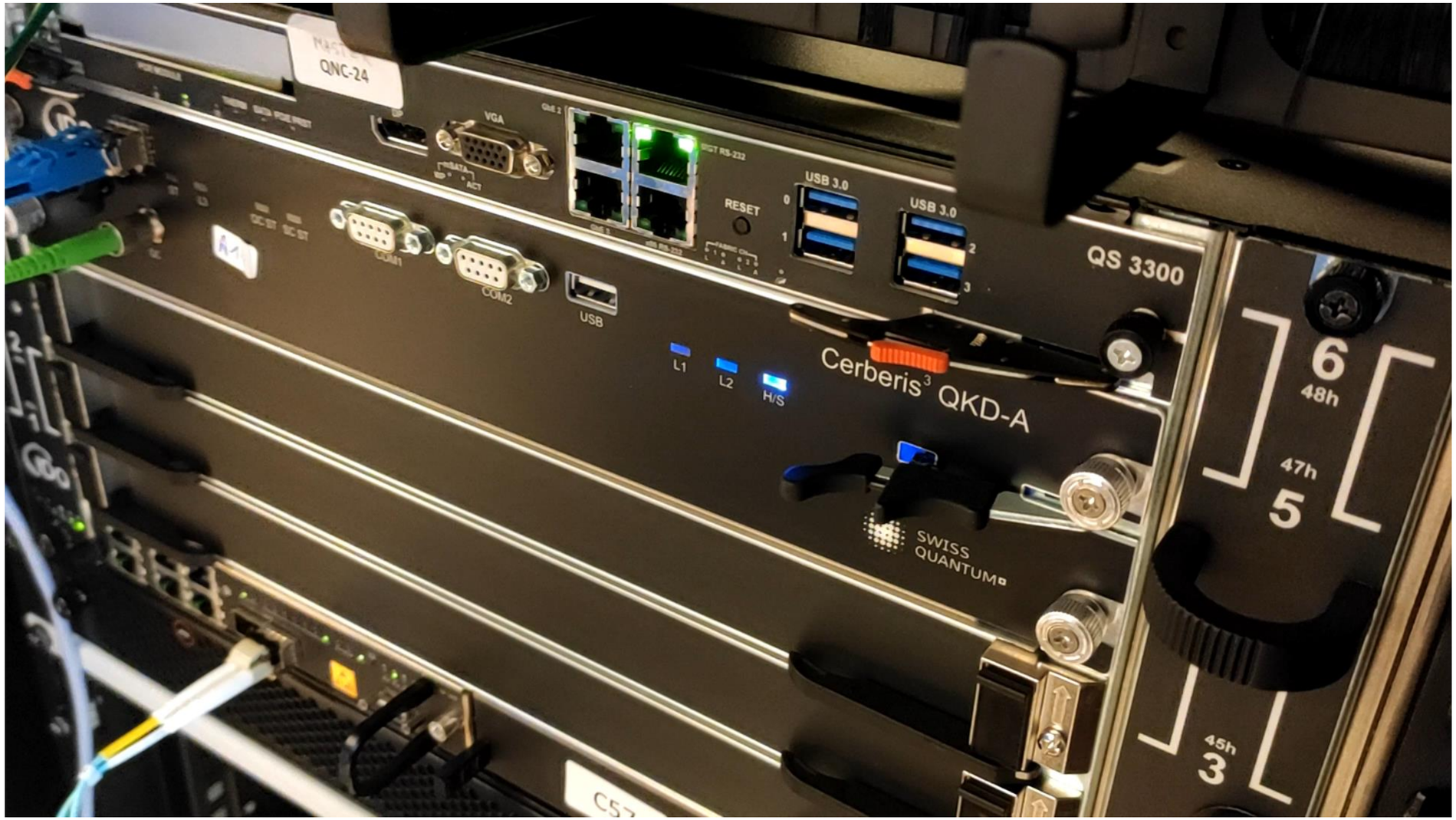
Cerberis 3 – QKD card insertion



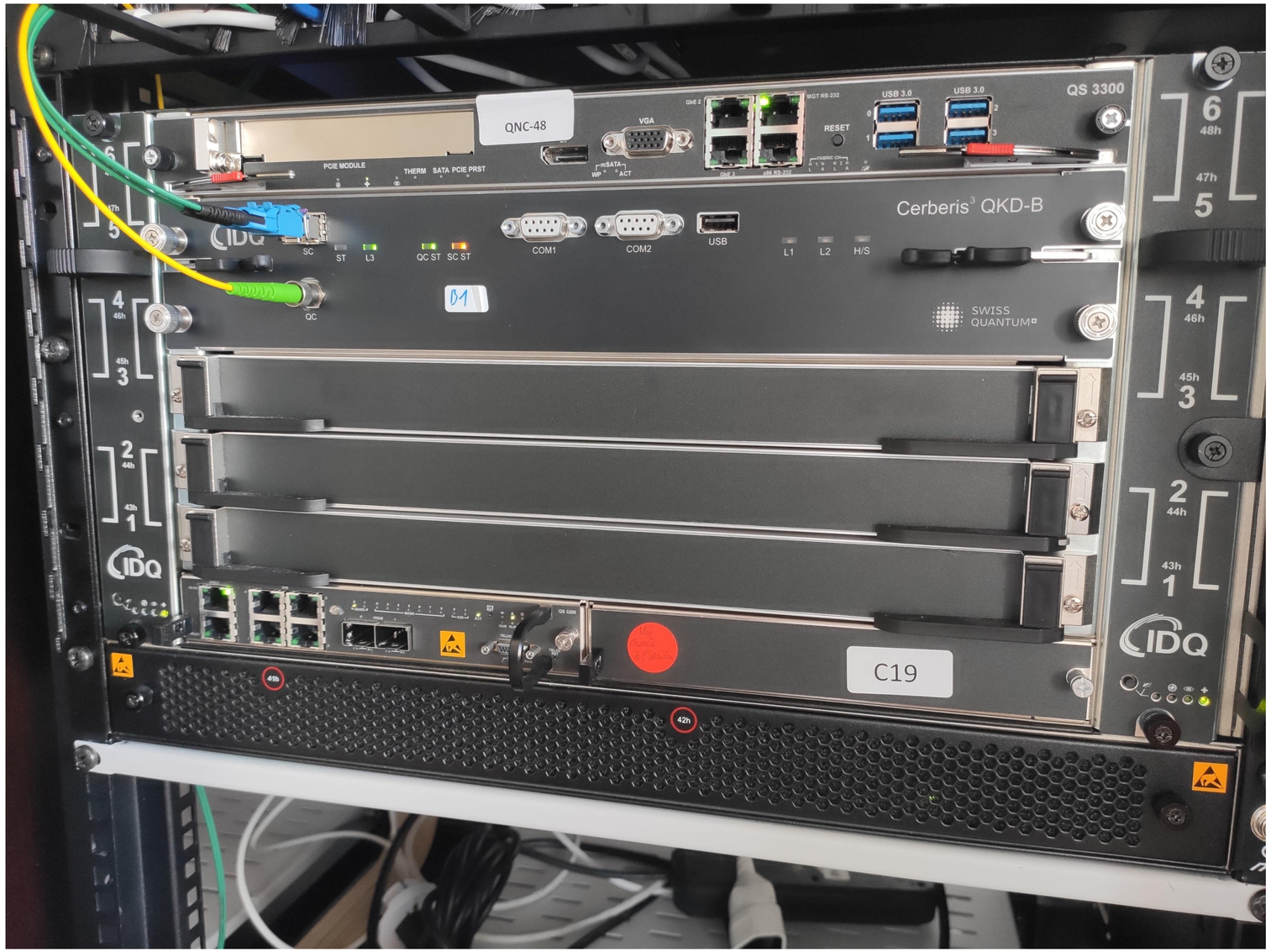
- Install QKD card as shown in the pictures
- Once the QKD card is inserted the **blue light should disappear**. If it's not the case you should check the bracket.
- Bracket are very sensitive, if they are not well positioned the QKD cards **will not start**.

Cerberis 3 – QKD card insertion

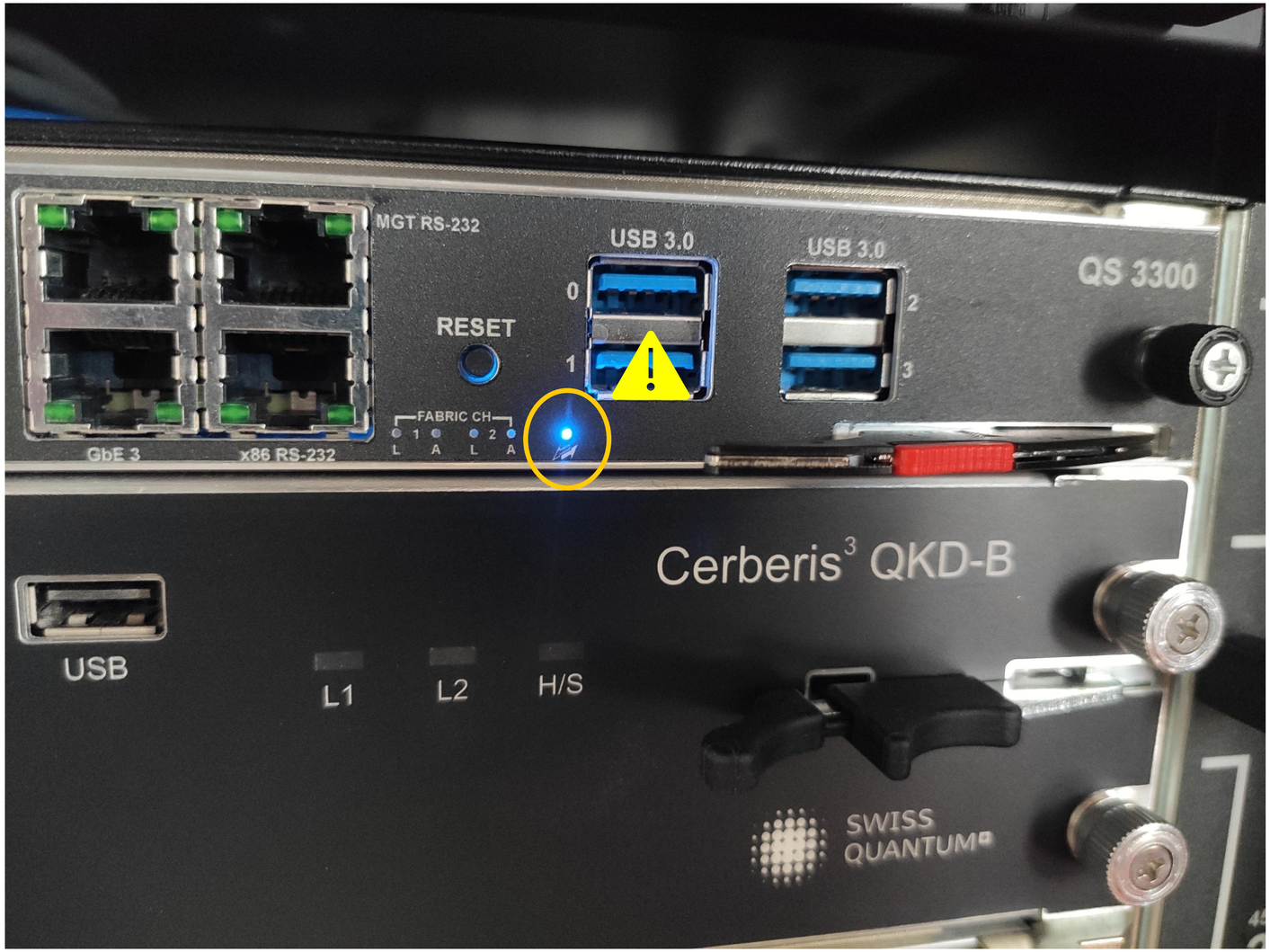
A video is much simpler to understand ☺



Cerberis 3 – QNC card insertion



Cerberis 3 – QNC card insertion



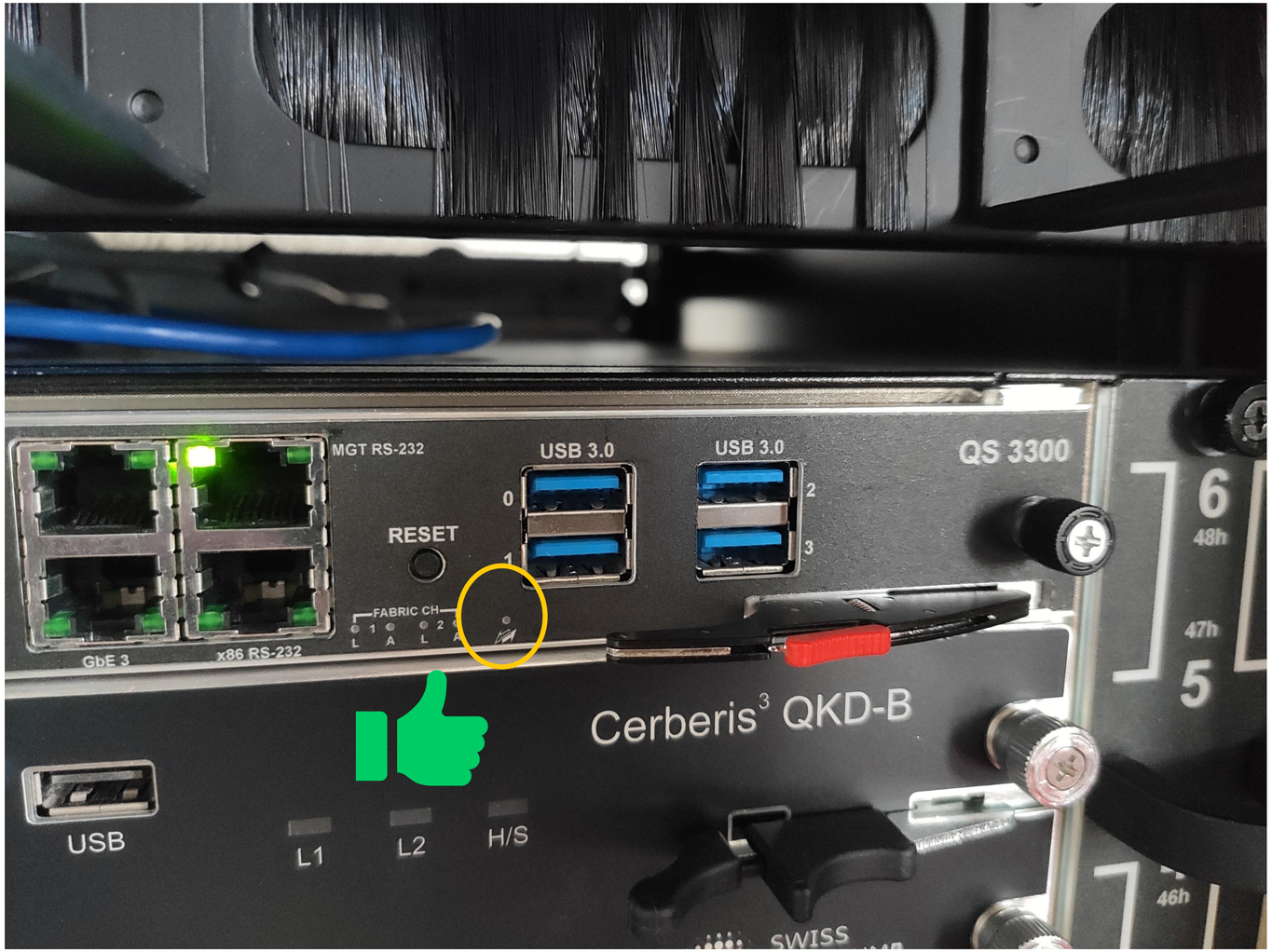
Not OK

Blue light is fixed and doesn't disappear

Cerberis 3 – QNC card insertion

Check if bracket is properly inserted

Cerberis 3 – QNC card insertion



Blue light disappeared

→ Card is started

Cerberis 3 – Physical setup (Quantum Channel)

Connect Quantum Channel

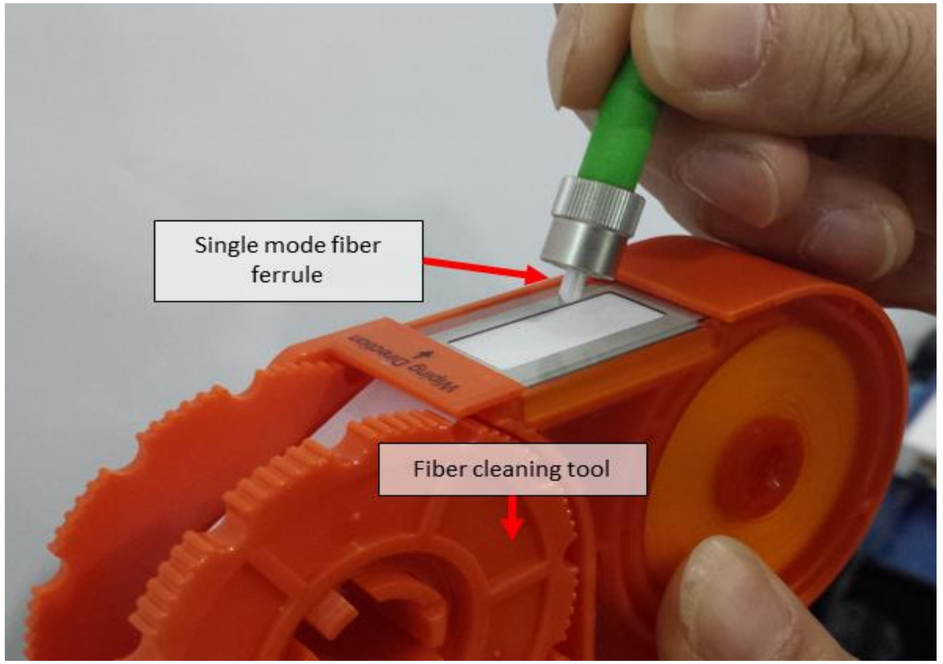
If you are using a short fiber
add fix attenuator
<u>on BOB side</u>

FC/FC APC SM fiber

+

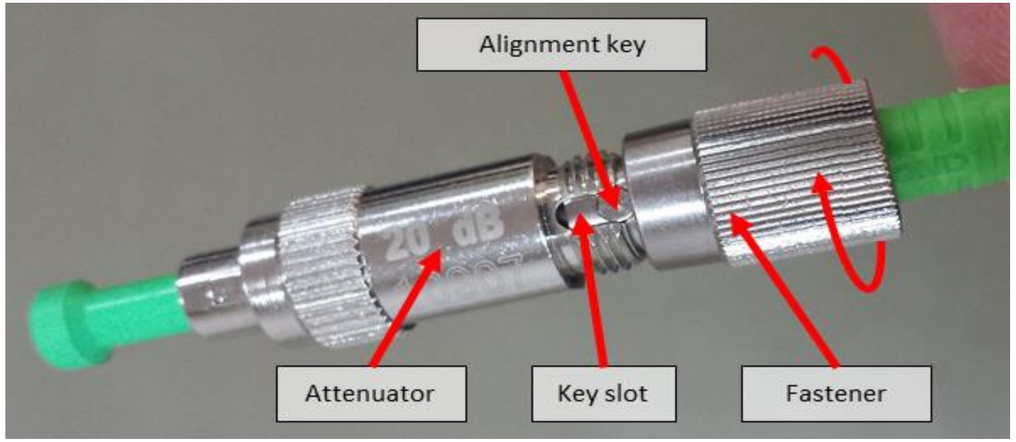
Attenuator (optional)

Cerberis 3 – Physical setup (Quantum Channel)



- Remove the protective cap from the FC/APC 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!

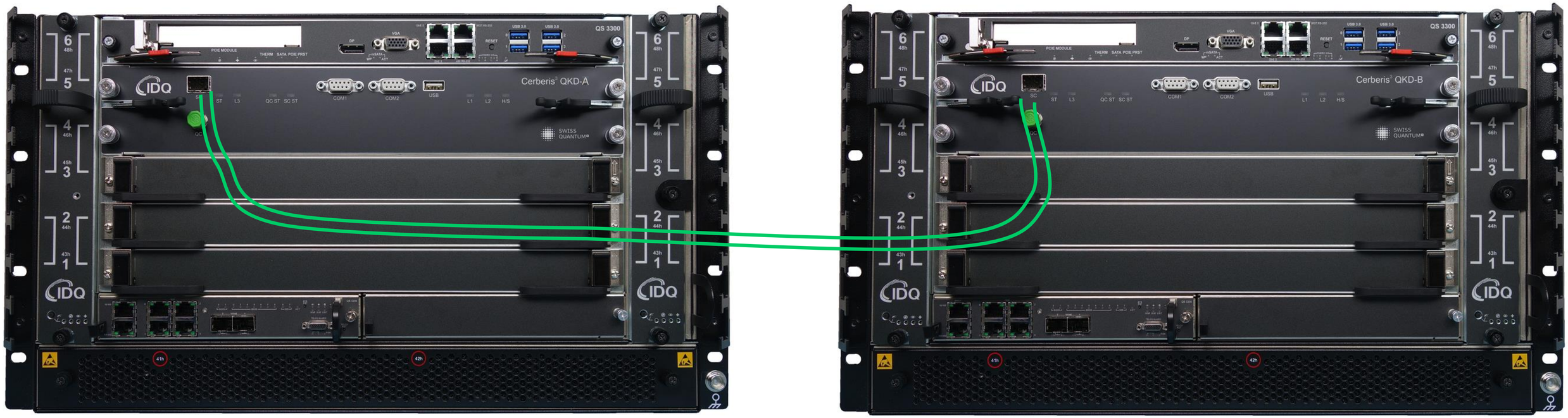
Cerberis 3 – Physical setup (Quantum Channel)



- Use a FC/**A**PC connector
- Connect the attenuator if needed in Bob's side
- Make sure the fiber is properly screwed into the port.

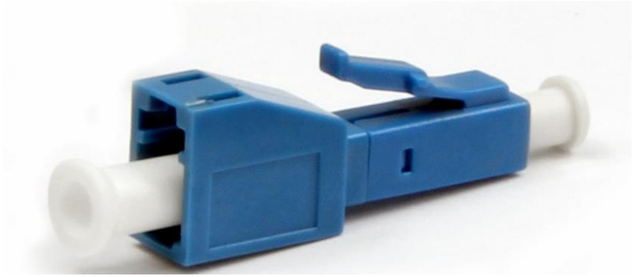
Cerberis 3 – Physical setup (Service Channel)

Connect Service Channel



LC/LC SMF fiber

+



Attenuator (optional)

Cerberis 3 – Physical setup (Service Channel)

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

Installation procedure – Power ON

SWISS QUANTUM

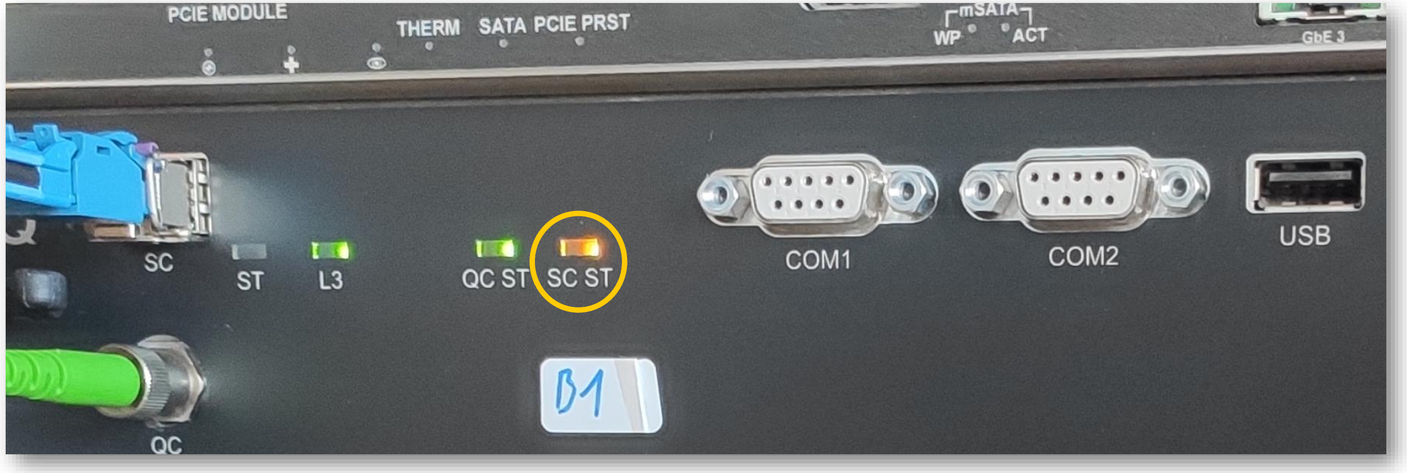
Details

- Note when QKD system is starting ALICE SC ST LED is blinking and BOB SC ST is not blinking.
- That means BOB is in the process of cooling down the detectors
- When QC ST (**Q**uantum **C**hannel **ST**atus) is green that means the Quantum Channel is working
- When SC ST (**S**ervice **C**hannel **St**atus) is yellow that means the service channel is aligned

IDQ

Installation procedure – Power ON

LED Service Channel



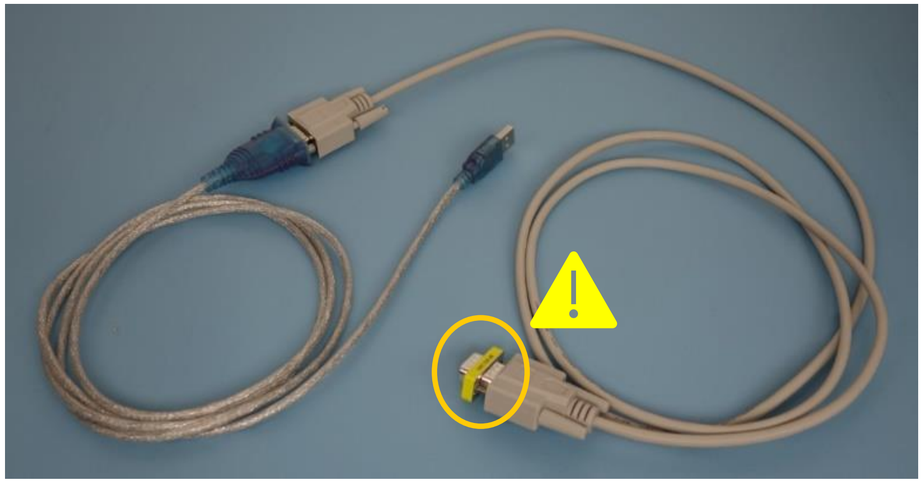
LED Status	Service Channel Status Description
ON (Yellow)	Service Channel active and synchronized.
Blinking (Yellow)	Service Channel synchronizing.
Blinking (Red)	Service Channel desynchronized.
OFF	Service Channel inactive.

Table 5: Description of the Service Channel Front Panel LED status.

Cerberis 3 – Console cables

The QKD is delivered with **2 types** of console cables.

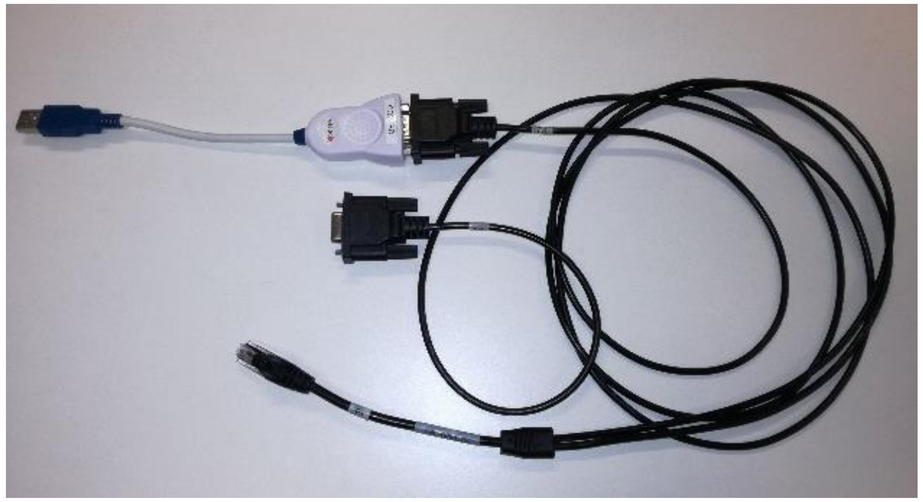
They are mandatory to do the basic configuration



DB9 cables + mini gender changer

Used to connect to:

- QKD card



Vadatech “Y cable”

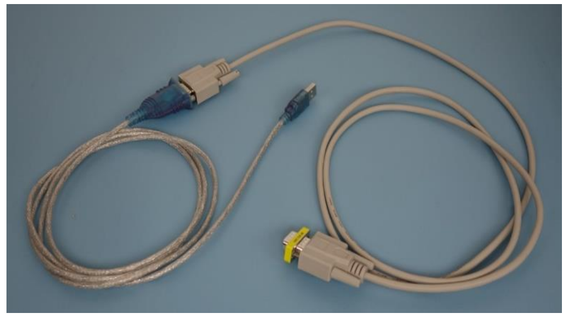
Necessary to connect to:

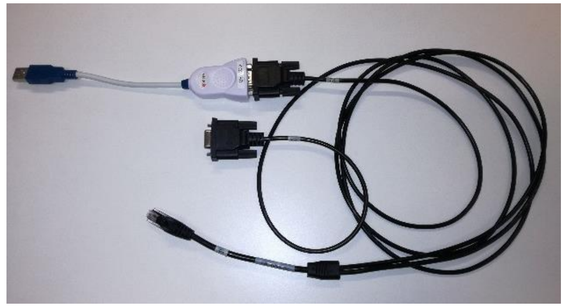
- QNC card

- Shelf Manager

Cerberis 3 – Physical setup (Console)

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





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

Configuration (CLI)

CERBERIS 3

Preliminary configuration

Cerberis 3 – Preliminary configuration

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 QNC card
- Be able to configure the QKD card
- Be able to configure the Shelf Manager

Cerberis 3 – Preliminary configuration

SWISS QUANTUM

Management:

- **All configuration** is done **through CLI** using
  - **SSH (port 22)**
  - **Serial Console connection**
- **There is no GUI** or graphical interfaces
- **Health and status** of the system can be checked using
  - **Logs**
  - **SNMPv3**
- We advise using **SNMP first**
- **Logs** should be used for **troubleshooting** only
- 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.

IDQ

Cerberis 3 – Preliminary configuration

SWISS QUANTUM

I. IP addresses Plan

- Understand the needed IP addresses
- Choose the needed IP addresses

II. SHMM IP addresses configuration

- Configure SHMM Bound IP address
- Configure SHMM Management IP address

III. QNC configuration

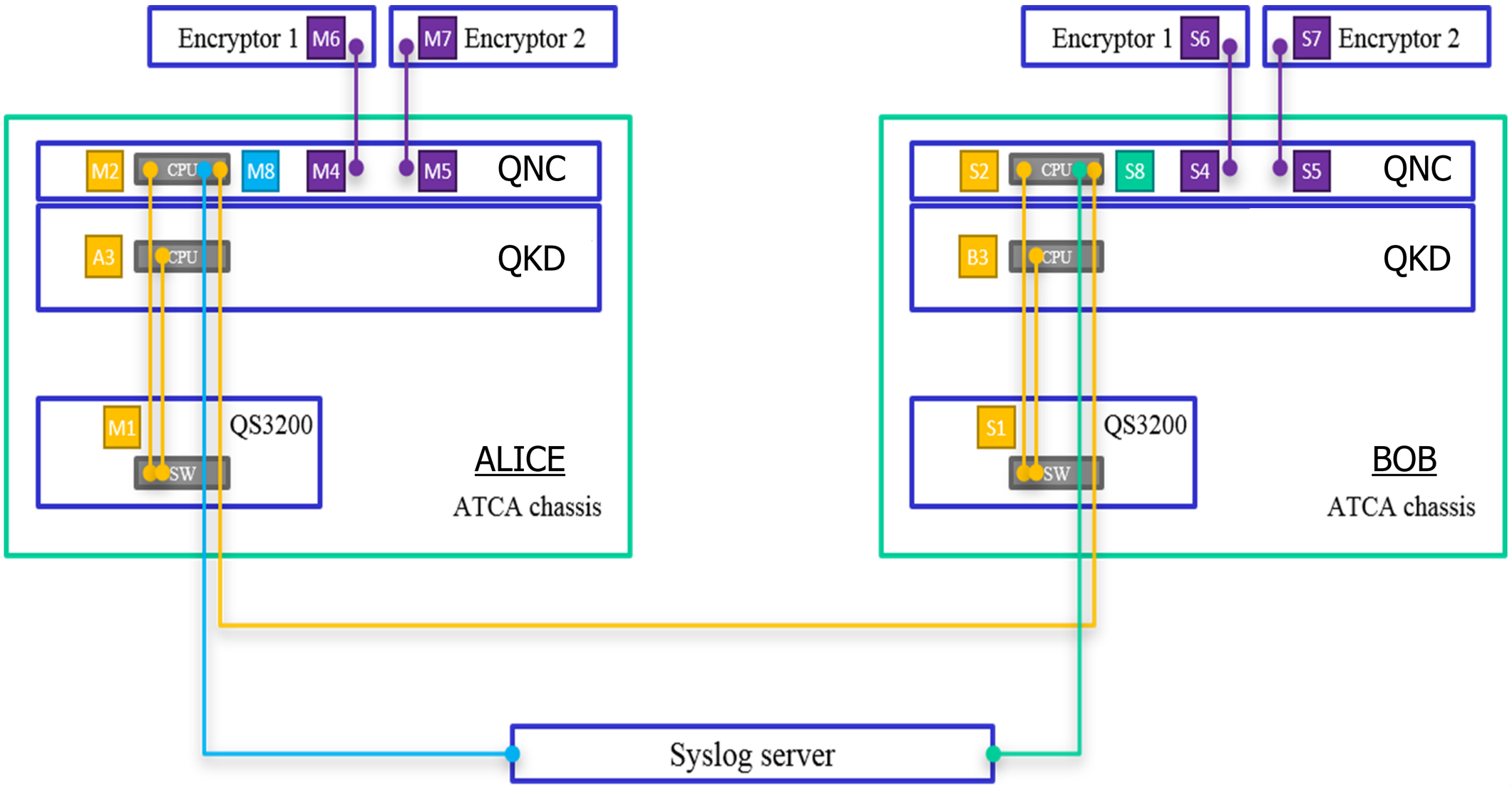
- Configure date and time
- Set the QNC Management IP address
- Configure frontpane IP addresses

IV. QKD configuration

- Configure date and time
- Set the QKD Management IP address
- Insert the IP address of the QNC on the QKD

IDQ

Cerberis 3 – Preliminary configuration



Cerberis 3 – Preliminary configuration

- **The Switch (M1/S1)**: This address corresponds to the Switch front panel ethernet ports GbE0-3 for Management (SNMP, syslog) and internode communications. It must be in same subnet as the QNC and QKD Blades.

- **The QNC-CPU (M2/S2)**: This address must be in same subnet as the QKD Blade and switch. It is used to connect to the QNC-CLI. It must also be given to the SNMP client to obtain the SNMP monitoring information. The local QKD Blades also need to be configured in order to know this IP address and to be able to communicate with it.

- **The QKD-CPU (A3/B3)**: This address must be in same subnet as the QNC and switch. It is used to connect to the QKD-CLI. The local QNC needs to be configured in order to know this IP address and to be able to communicate with it.

Cerberis 3 – Preliminary configuration

- **The QNC key interface to the encryptors implementing ETSI interface (M4+M5/S4+S5)**: This address is used to request keys from the QNC and must be given to the encryptors that request the keys from the QNC. It must therefore be in same subnet as the encryptor.

- **The interfaces of the encryptors implementing ETSI interface (M6+M7/S6+S7)**: This is the address of the encryptors that request keys from the QNC. It must therefore be in same subnet as the QNC-Key interface (GbE2).

- **The syslog server (M8/S8)**: This address must be given to the syslog client for receiving alarms. The address must in the same subnet as QNC and switch.

- **The Shelf-Manager BMC (M9/S9)**: This address is only required when updating the BMC via the shelf-manager or switch.

Cerberis 3 – Preliminary configuration

SWISS QUANTUM

I. IP addresses Plan

- Understand the needed IP addresses
- Choose the needed IP addresses

II. SHMM IP addresses configuration

- Configure SHMM Bound IP address
- Configure SHMM Management IP address

III. QNC configuration

- Configure date and time
- Set the QNC Management IP address
- Configure frontpane IP addresses

IV. QKD configuration

- Configure date and time
- Set the QKD Management IP address
- Insert the IP address of the QNC on the QKD

IDQ

SWISS QUANTUM

LAB IP addresses

10.10.xxx.xxx KEMS 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

IDQ

Cerberis 3 – Preliminary configuration

SWISS QUANTUM

I. IP addresses Plan

- Understand the needed IP addresses
- Choose the needed IP addresses

II. SHMM IP addresses configuration

- Configure SHMM Bound IP address
- Configure SHMM Management IP address

III. QNC configuration

- Configure date and time
- Set the QNC Management IP address
- Configure frontpane IP addresses

IV. QKD configuration

- Configure date and time
- Set the QKD Management IP address
- Insert the IP address of the QNC on the QKD

IDQ

Cerberis 3 – Preliminary configuration

SHMM:

- **2 IP addresses** must be **configured**:
  - ❑ Bound IP address (used for updates)
  - ❑ Management IP address (use for normal management)

Cerberis 3 – Preliminary configuration

SWISS QUANTUM

I. IP addresses Plan

- Understand the needed IP addresses
- Choose the needed IP addresses

II. SHMM IP addresses configuration

- Configure SHMM Bound IP address
- Configure SHMM Management IP address

III. QNC configuration

- Configure date and time
- Set the QNC Management IP address
- Configure frontpane IP addresses

IV. QKD configuration

- Configure date and time
- Set the QKD Management IP address
- Insert the IP address of the QNC on the QKD

IDQ

Cerberis 3 – Preliminary configuration

1. Connect to the QS3200 through the RS-232 port

2. Type
   set_ip_connection -i 0 -a <ShMM_IP_Addr> -g <ShMM_gateway> -n <ShMM_netmask>

3. Press enter

4. Enter reboot

5. Connect to the QS3200 through the RS-232 port

6. Type get_ip_connection and press enter

7. The current configuration is showing

Cerberis 3 – Preliminary configuration

SWISS QUANTUM

I. IP addresses Plan

- Understand the needed IP addresses
- Choose the needed IP addresses

II. SHMM IP addresses configuration

- Configure SHMM Bound IP address
- Configure SHMM Management IP address

III. QNC configuration

- Configure date and time
- Set the QNC Management IP address
- Configure frontpane IP addresses

IV. QKD configuration

- Configure date and time
- Set the QKD Management IP address
- Insert the IP address of the QNC on the QKD

IDQ

Cerberis 3 – Preliminary configuration

1. Connect to the QS3200 through the RS-232 as described in section 5.2.5.
2. In the file /etc/rc.d/rc.conf, edit the interface 4 parameters using the installed vieditor:
3. Enter vi /etc/rc.d/rc.conf
4. Press i to be in Edit mode
5. Set the settings corresponding to item M1/S1 in Table 2, section 4.3.2.:
   - Set HOSTNAME value to <switch_hostname>
   - Set IPADDR4 value to <switch_IP_address>
   - Set NETASK4 value to <switch_netmask>
   - Remove the BROADCAST4 line

Cerberis 3 – Preliminary configuration

6. Save and close the interface configuration file:
7. Type :wq
8. Press Enter.
9. Reboot the QS3200:
10. Type reboot and press Enter
11. Check if you can ping the IP address and connect to it using SSH

Cerberis 3 – Preliminary configuration

SWISS QUANTUM

I. IP addresses Plan

- Understand the needed IP addresses
- Choose the needed IP addresses

II. SHMM IP addresses configuration

- Configure SHMM Bound IP address
- Configure SHMM Management IP address

III. QNC configuration

- Configure date and time
- Set the QNC Management IP address
- Configure frontpane IP addresses

IV. QKD configuration

- Configure date and time
- Set the QKD Management IP address
- Insert the IP address of the QNC on the QKD

IDQ

Cerberis3 – 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
#?
```

First login as “admin”

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 2021-02-09T12:03:00
2021-02-09 12:03:00 -05:00
Tue Feb  9 12:03:01 EST 2021
admin>
```

Command to configure date and time

Cerberis 3 – Preliminary configuration

SWISS QUANTUM

I. IP addresses Plan

- Understand the needed IP addresses
- Choose the needed IP addresses

II. SHMM IP addresses configuration

- Configure SHMM Bound IP address
- Configure SHMM Management IP address

III. QNC configuration

- Configure date and time
- Set the QNC Management IP address
- Configure frontpane IP addresses

IV. QKD configuration

- Configure date and time
- Set the QKD Management IP address
- Insert the IP address of the QNC on the QKD

IDQ

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 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]
```

You have to specify the IP address of the QNC

Help

Set QNC management IP address

Cerberis 3 – Preliminary configuration

You have to specify the IP address of the QNC

```
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 Frontpane Key Delivery Interfaces

Cerberis 3 – Preliminary configuration

SWISS QUANTUM

I. IP addresses Plan

- Understand the needed IP addresses
- Choose the needed IP addresses

II. SHMM IP addresses configuration

- Configure SHMM Bound IP address
- Configure SHMM Management IP address

III. QNC configuration

- Configure date and time
- Set the QNC Management IP address
- Insert the IP address of the SHMM on the QNC
- Configure frontpane IP addresses

IV. QKD configuration

- Configure date and time
- Set the QKD Management IP address
- Insert the IP address of the QNC on the QKD

IDQ

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 or change the host name
  test                    test network

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

Help and command's information

```
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]
```

Command to use with arguments

Configure the Frontpane port using the "network" command:

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

IDQ

Cerberis 3 – Preliminary configuration

SWISS QUANTUM

I. IP addresses Plan

- Understand the needed IP addresses
- Choose the needed IP addresses

II. SHMM IP addresses configuration

- Configure SHMM Bound IP address
- Configure SHMM Management IP address

III. QNC configuration

- Configure date and time
- Set the QNC Management IP address
- Insert the IP address of the SHMM on the QNC
- Configure frontpane IP addresses

IV. QKD configuration

- Configure date and time
- Set the QKD Management IP address
- Insert the IP address of the QNC on the QKD

IDQ

Cerberis 3 – 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 3 – Preliminary configuration

SWISS QUANTUM

I. IP addresses Plan

- Understand the needed IP addresses
- Choose the needed IP addresses

II. SHMM IP addresses configuration

- Configure SHMM Bound IP address
- Configure SHMM Management IP address

III. QNC configuration

- Configure date and time
- Set the QNC Management IP address
- Insert the IP address of the SHMM on the QNC
- Configure frontpane IP addresses

IV. QKD configuration

- Configure date and time
- Set the QKD Management IP address
- Insert the IP address of the QNC on the QKD

IDQ

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 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]
```

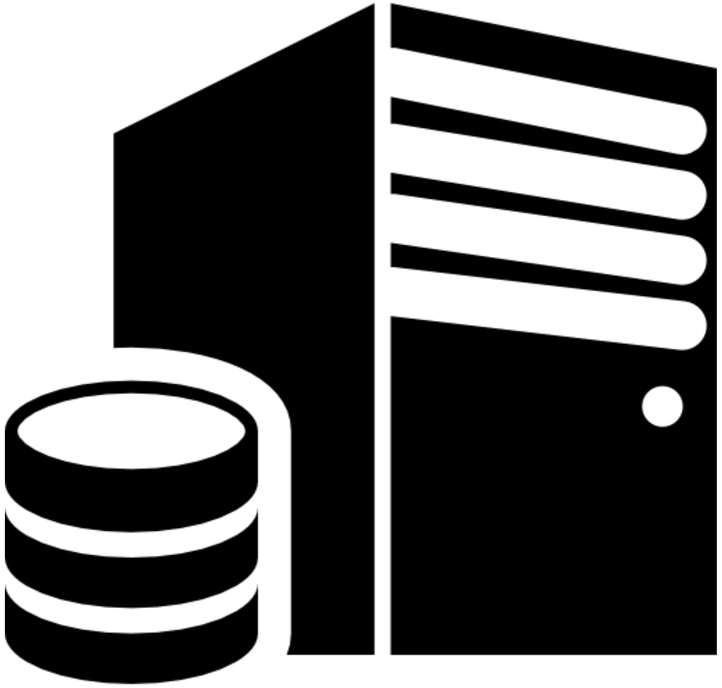
Admin role must be used for these changes

Help and command's information

Command to use with arguments

Type "network" to confirm changes

KEMS installation



KEMS installation

Docker installation

- Follow the installation instructions provided by Docker
- Currently KEMS is only supported on CentOS 7 (Not CentOS 8): https://docs.docker.com/install/linux/docker-ce/centos/
  Recommended to install latest version of docker – docker version included in CentOS7 may not be sufficient.

KEMS installation

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

- *tar xvf KEMS-x.y.z-ttt.tar.gz*
- *mv kems-config-api/kems-config-api-<version> .*
- *rmdir kems-config-api*

This will create a new folder containing these files:
3 docker images packed in a single file, **docker-images.tar**:
KEMS database docker image
KEMS server docker image
IDQ QN ConfigWebApi docker image
1 configuration file ***docker-compose.yml***
1 configuration file ***docker-compose.overrride.yml***

KEMS installation

Use this command to load the images into docker:
docker load -i kems-config-api-<version>/docker-images.tar

Go inside the created folder:
cd kems-config-api-<version>

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 to include your proper time zone

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

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

KEMS installation

Docker installation

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

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

- **To check if the both containers are well started**

```
/home/idq/KEMS_2.1.3/KEMS$ sudo docker ps -a
CONTAINER ID   IMAGE                 COMMAND                  CREATED          STATUS          PORTS                    NAMES
c7211a2bd546   kems-database:latest  "docker-entrypoint.s…"   28 seconds ago   Up 24 seconds   3306/tcp                 kems_kems-database_1
1cfdd2bd6ff6   kems-server:latest    "./wait-for-it.sh ke…"   28 seconds ago   Up 25 seconds   0.0.0.0:8080->8080/tcp   kems_kems-server_1
/home/idq/KEMS_2.1.3/KEMS$
```

- **To check if the both containers are well started**

/home/idq/KEMS_2.1.3/KEMS$ **sudo docker logs 1cfdd2bd6ff6**

KEMS installation

Docker installation

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

- **disable SE Linux**

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

- **disable the firewall**

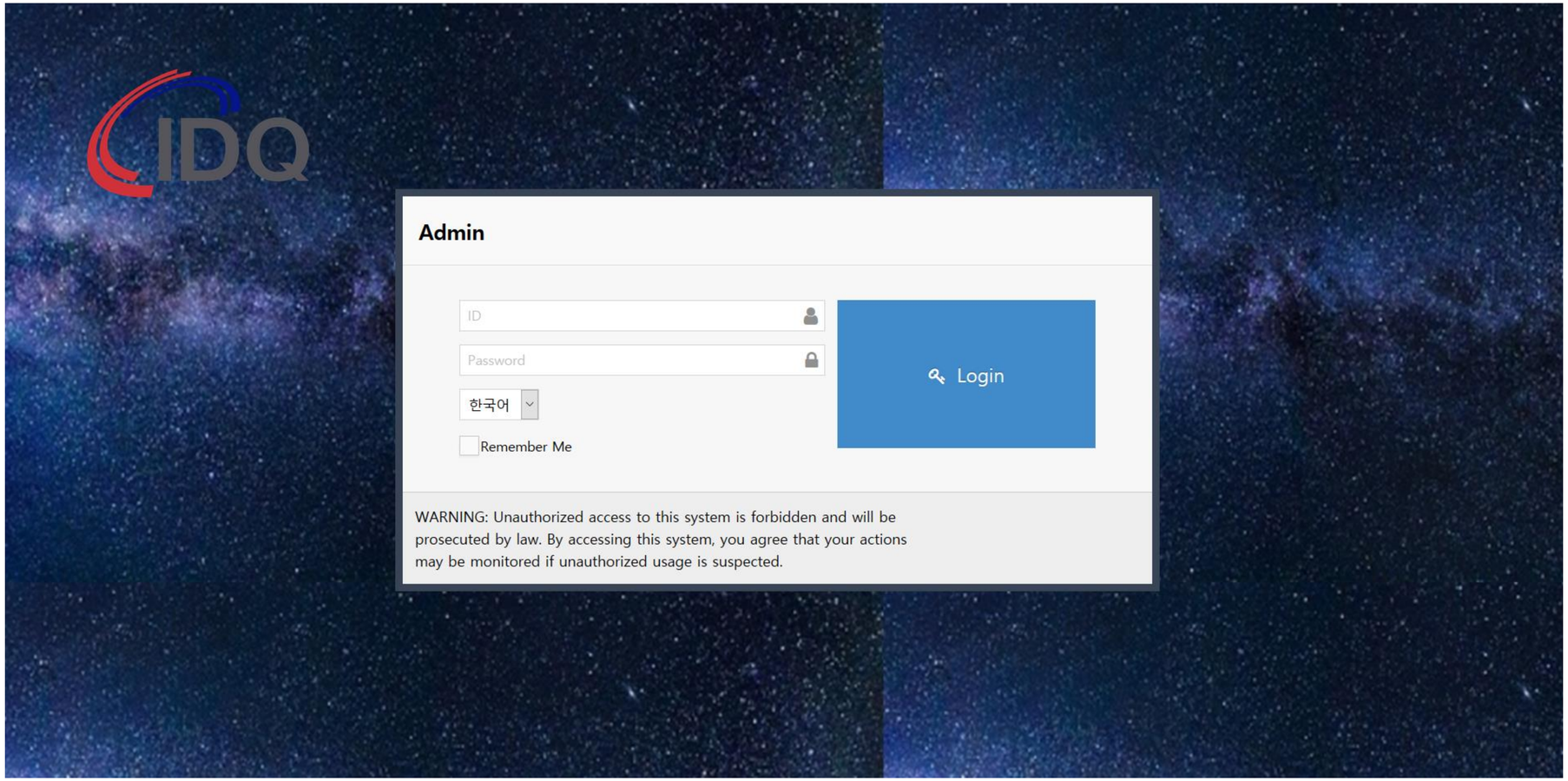
```
# systemctl stop firewalld
# systemctl disable firewalld
```

KEMS configuration

KEMS installation

Access to the GUI

- **You should be able to access the webpage** (http://x.x.x.x:8080)
- This now redirects to secure port 9443 (or you can access directly at x.x.x.x:9443)
- **Use Admin / Admin**



KEMS installation

Known Limitations

- In the KEMS GUI, during the configuration, the Consumer, Provider, KMS and links name are limited to **16 characters maximum.**

- The maximum keys per request is 16 keys.

- **Do not trust equipment's color** in the KEMS GUI; **A KMS in red color and in alarm could work properly.**

- Only the "Any" network type is used for all configurations in KEMS.

KEMS installation

Naming on KEMS

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

  - **KMS** → QNC
  - **Provider** → QKD
  - **Consumer** → Encryptor
  - **KMS link** → Link between QNC
  - **Provider link** → Link between QKD
  - **Consumer link** → Link between encryptors

KEMS processes

SWISS QUANTUM

I. Preliminary configuration

- Create a group on KEMS

II. QNC configuration

- Set the QNC name on each QNC and write down the UID
- Create each QNC on the KEMS
- Check if the KEMS server is connected to each QNC
- Create the QNC link between both QNCs

III. QKD configuration

- Set each QKD on the KEMS
- Create the QKD link between the both QKDs

IDQ

KEMS processes

SWISS QUANTUM

IV. Encryptors configuration

- Create the encryptors on KEMS
- Create the link between the encryptors
- Add certificates on encryptors

V. Trusted path

- Create the trusted path
- Deploy the trusted path

VI. Checking

- Check if the system is properly working and exchanging keys

IDQ

KEMS processes

I. Preliminary configuration

- Create a group on KEMS

II. QNC configuration

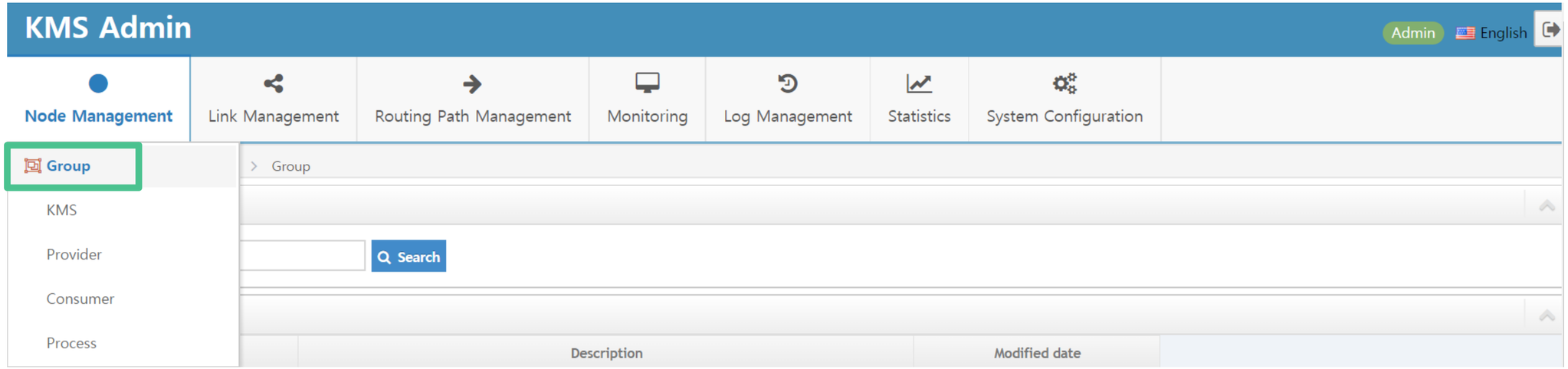
- Set the QNC name on each QNC and write down the UID
- Create each QNC on the KEMS
- Check if the KEMS server is connected to each QNC
- Create the QNC link between the both QNCs

III. QKD configuration

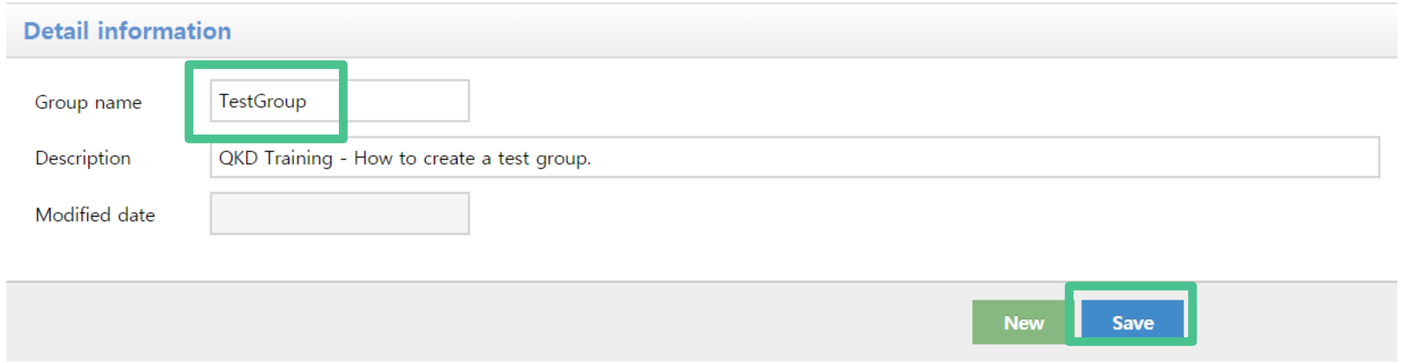
- Set each QKD on the KEMS
- Create the QKD link between both QKDs

KEMS configuration

- To add a group first in KEMS (execute these steps if you add KMS I unexacting Group):
  Connect to the KEMS GUI
  Create a Group: Node Management / Group / Group name



- Enter group name and description then click save.



KEMS processes

SWISS QUANTUM

I. Preliminary configuration

- Create a group on KEMS

II. QNC configuration

- Set the QNC name on each QNC and write down the UID
- Create each QNC on the KEMS
- Check if the KEMS server is connected to each QNC
- Create the QNC link between the both QNCs

III. QKD configuration

- Set each QKD on the KEMS
- Create the QKD link between both QKDs

IDQ

KEMS configuration

- 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**

KEMS processes

SWISS QUANTUM

I. Preliminary configuration

- Create a group on KEMS

II. QNC configuration

- Set the QNC name on each QNC and write down the UID
- Create each QNC on the KEMS
- Check if the KEMS server is connected to each QNC
- Create the QNC link between the both QNCs

III. QKD configuration

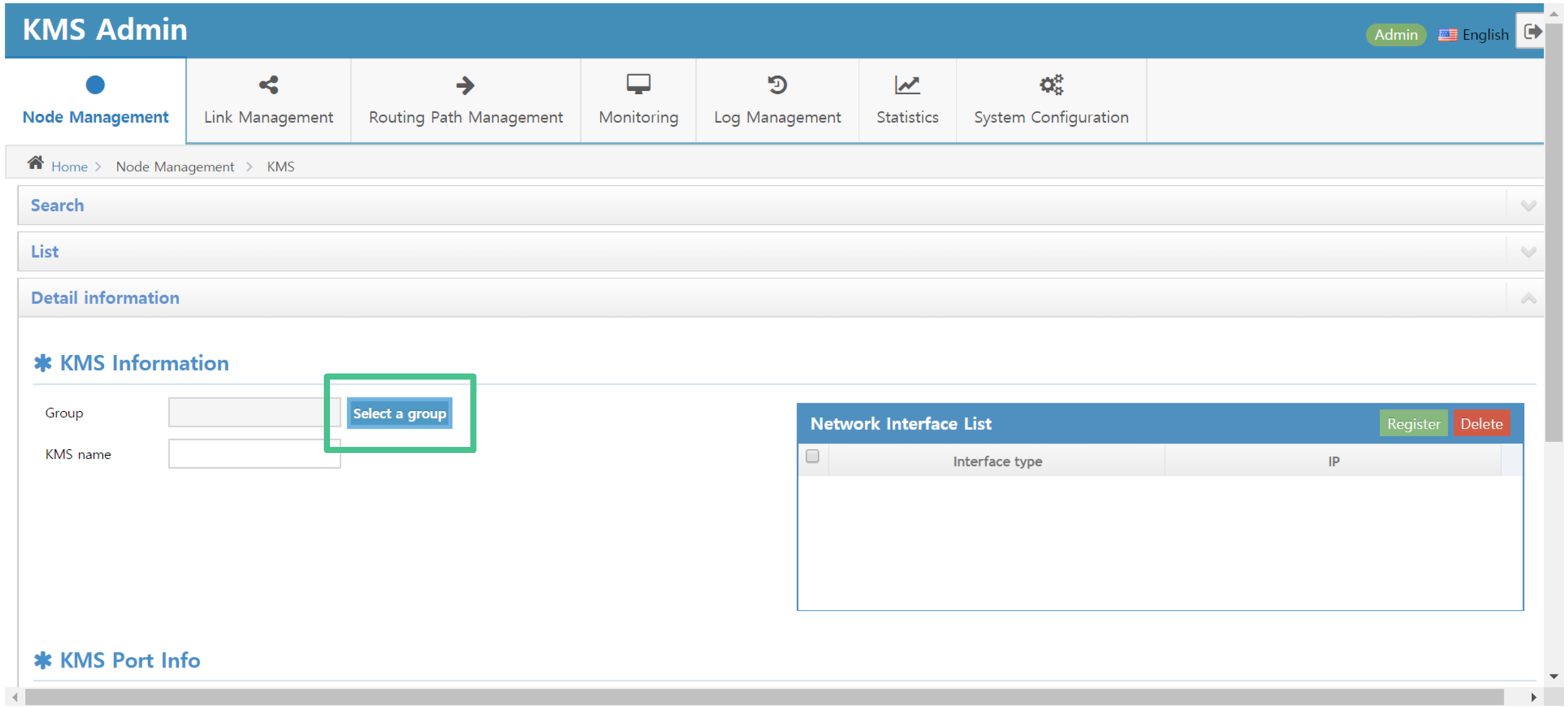
- Set each QKD on the KEMS
- Create the QKD link between both QKDs

IDQ

KEMS configuration

- Go to KEMS Web-apps, Node Management / KMS
- Create KMS and set the following fields (see screenshots below)

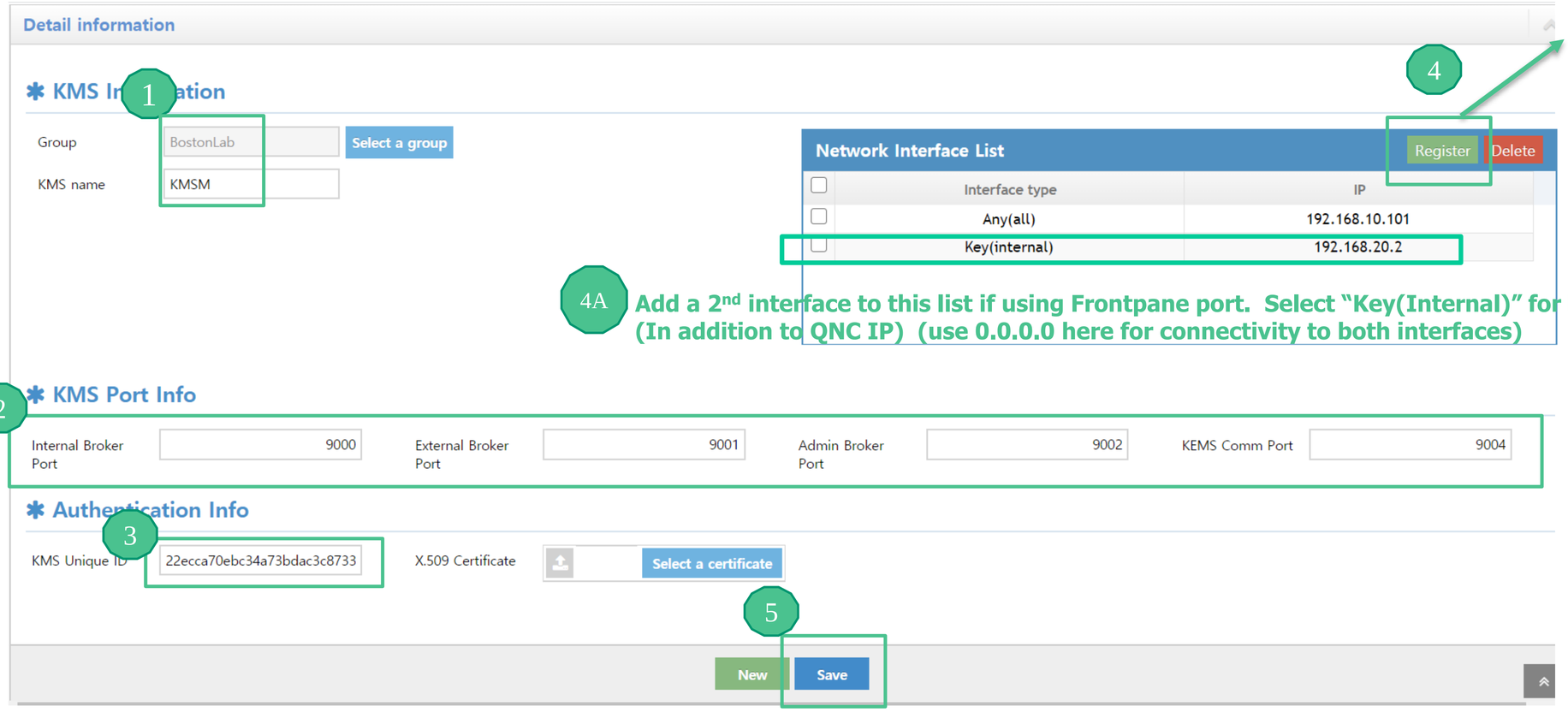
Configure 2x: 1 for each QNC

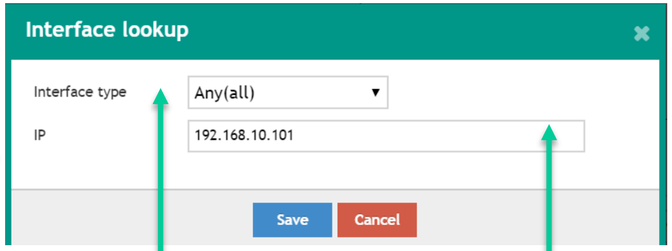


KEMS configuration

- The “KMS Name” is the same set in QNC and the “UID” is also the same in QNC.
- Create KMS and set the following fields (see screenshots below)

Configure 2x:
1 for each QNC





Select “Any(all)”

QNC IP

KEMS processes

SWISS QUANTUM

I. Preliminary configuration

- Create a group on KEMS
- Set the KEMS server IP address on each QNC

II. QNC configuration

- Set the QNC name on each QNC and write down the UID
- Create each QNC on the KEMS
- Check if the KEMS server is connected to each QNC
- Create the QNC link between the both QNCs

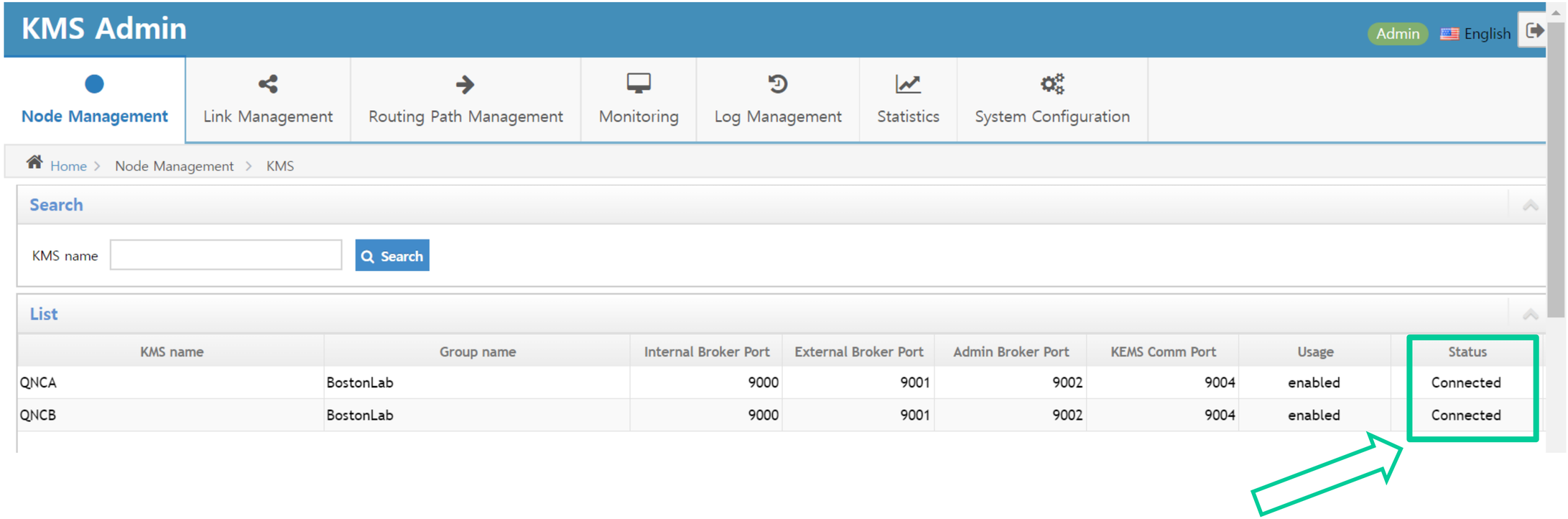
III. QKD configuration

- Set each QKD on the KEMS
- Create the QKD link between both QKDs

IDQ

KEMS configuration

Verify KEMS is communicating with each QNC:



KMS name	Group name	Internal Broker Port	External Broker Port	Admin Broker Port	KEMS Comm Port	Usage	Status
QNCA	BostonLab	9000	9001	9002	9004	enabled	Connected
QNCB	BostonLab	9000	9001	9002	9004	enabled	Connected

KEMS processes

SWISS QUANTUM

I. Preliminary configuration

- Create a group on KEMS
- Set the KEMS server IP address on each QNC

II. QNC configuration

- Set the QNC name on each QNC and write down the UID
- Create each QNC on the KEMS
- Check if the KEMS server is connected to each QNC
- Create the QNC link between the both QNCs

III. QKD configuration

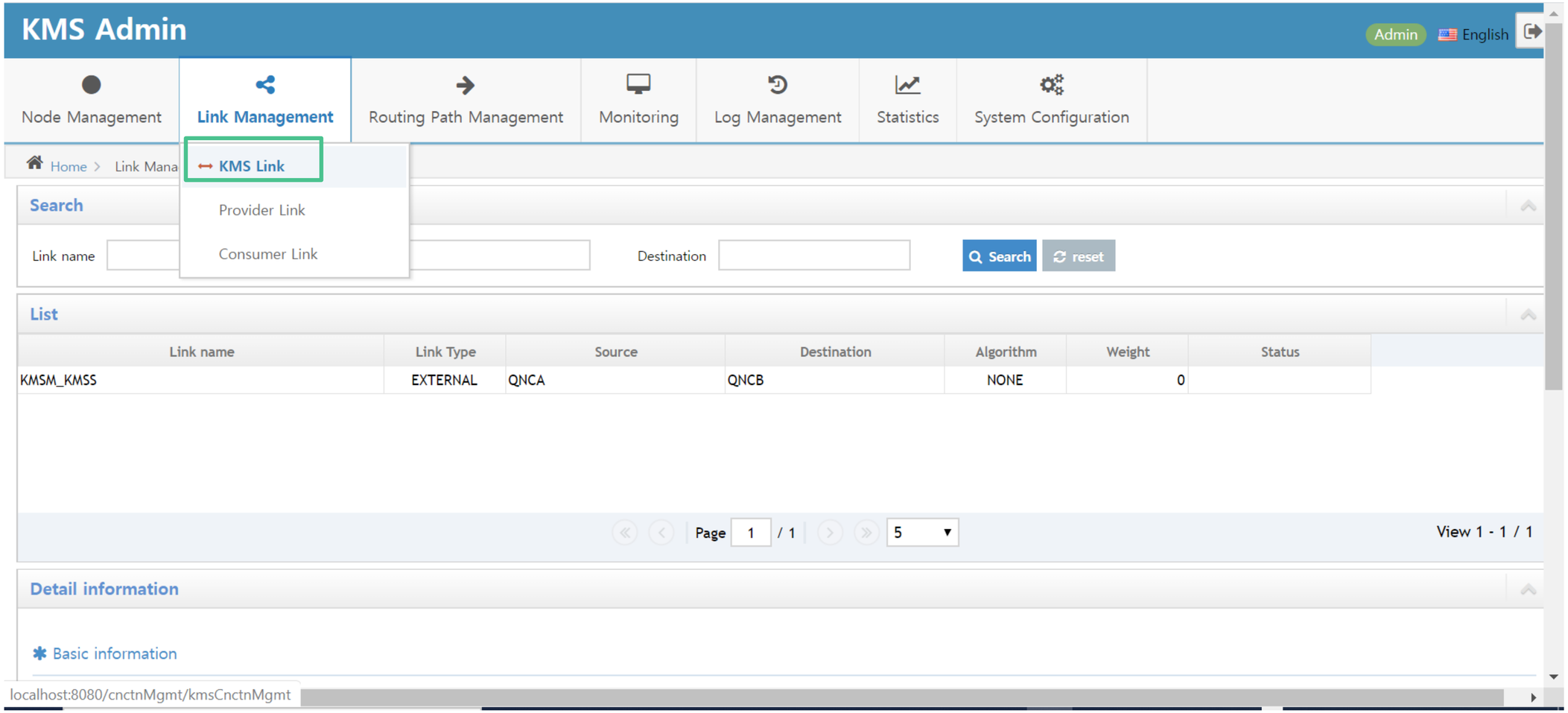
- Set each QKD on the KEMS
- Create the QKD link between both QKDs

IDQ

KEMS configuration

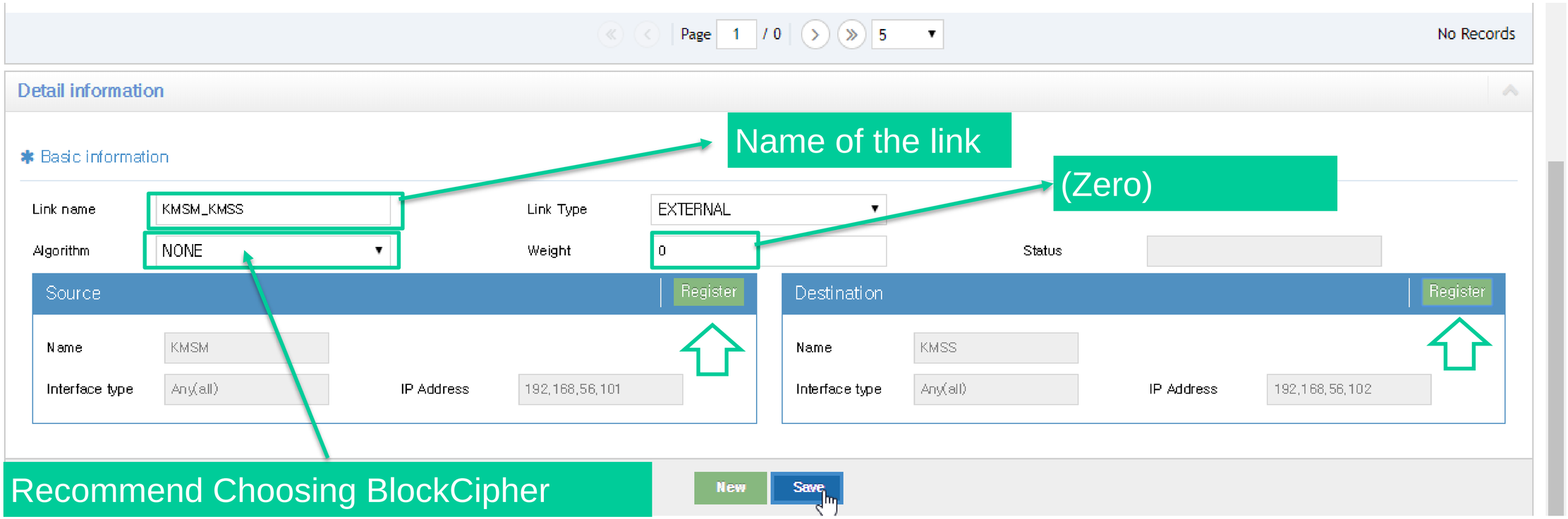
- Go to Link Management / KMS Link

- Create new link and set the following fields (see screenshots below):



KEMS configuration

- You need to add ALICE and BOB



- Leave the default parameters:

  Link name: Any identifier to name a link between master and slave KMS

  Link Type: External (fixed)

  Algorithm : BlockCipher or OTP (If using Trusted Node, Algorithm recommended to be Aria256 or OTP for security reasons)

  Weight : 0

KEMS processes

SWISS QUANTUM

I. Preliminary configuration

- Create a group on KEMS
- Set the KEMS server IP address on each QNC

II. QNC configuration

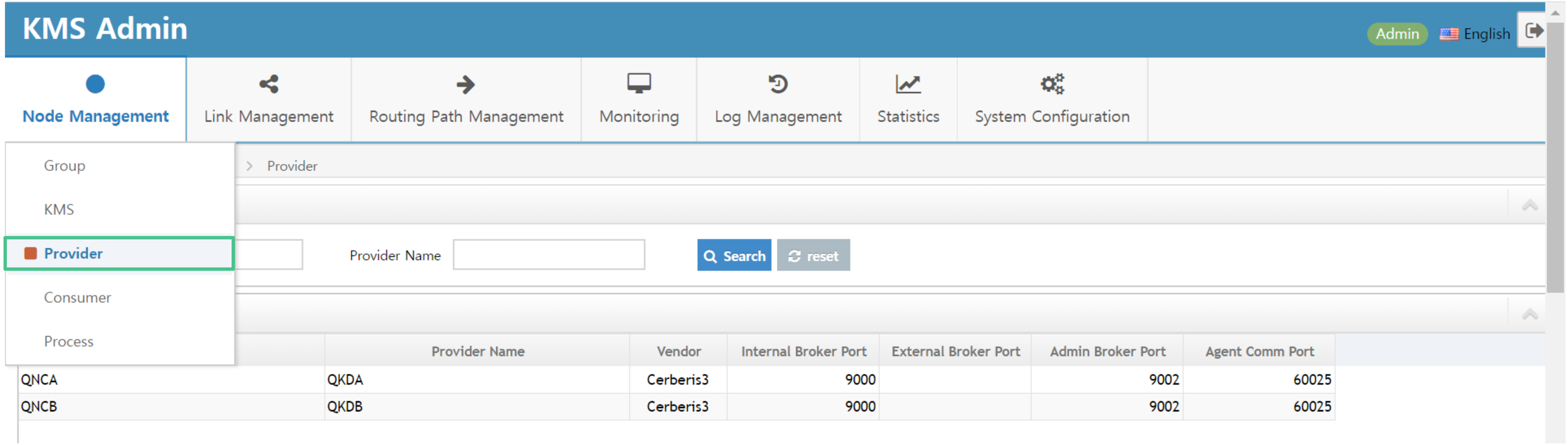
- Set the QNC name on each QNC and write down the UID
- Create each QNC on the KEMS
- Check if the KEMS server is connected to each QNC
- Create the QNC link between the both QNCs

III. QKD configuration

- Set each QKD on the KEMS
- Create the QKD link between both QKDs

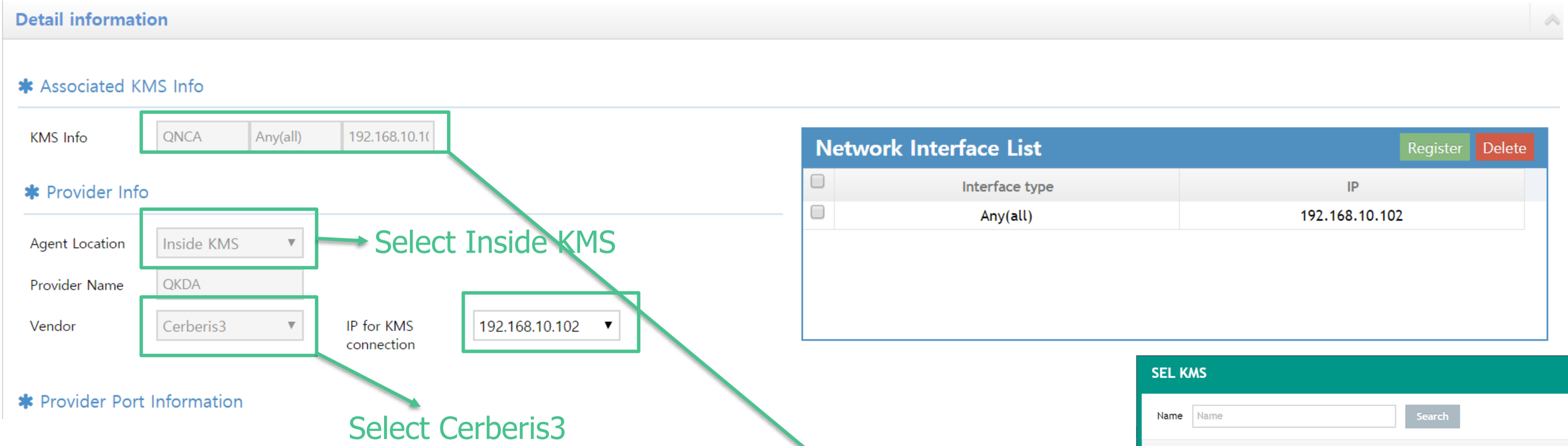
IDQ

KEMS configuration



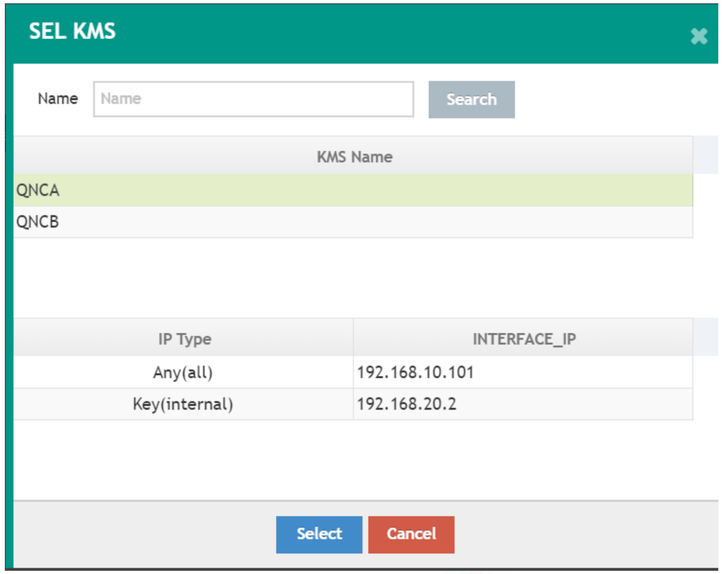
Configure 2x:
One for each QKD

KEMS configuration



Select Inside KMS

Select Cerberis3



Insert IP address of QNC Mgmt or Frontpane (Depending on which port will get key request)

Select Any (all) for Mgmt
Select Key (Internal) for Frontpane

Configure 2x:
One for each QKD

KEMS configuration

SWISS QUANTUM

Detail information

Associated KMS Info

KMS Info: QNCA | Any(all) | 192.168.10.1(| Select

Provider Info

Agent Location: Inside KMS

Provider Name: QNCA

Vendor: Cerberis3

IP for KMS connection: Select (4)

Select QKD IP

Provider Port Information

Key channel port: 5560

Management channel port: 5561

Signal channel port: 5562

New | Save (5)

Network Interface List — Register (1) | Delete

	Interface type	IP
☐	Any(all)	192.168.10.102

Register network interface

Interface type (2): Any(all) — Select Any(all)

IP (3): 192.168.10.102 — QKD IP

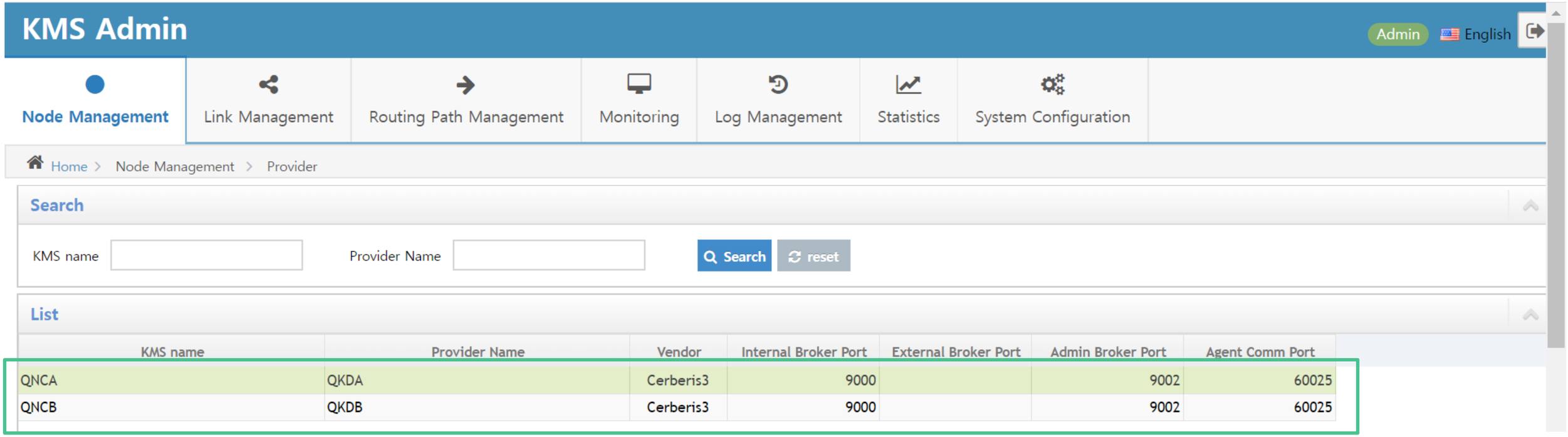
Save | Cancel

Configure 2x:
One for each QKD

IDQ

KEMS configuration

- Confirm the newly added providers in the list.



KMS name	Provider Name	Vendor	Internal Broker Port	External Broker Port	Admin Broker Port	Agent Comm Port
QNCA	QKDA	Cerberis3	9000		9002	60025
QNCB	QKDB	Cerberis3	9000		9002	60025

KEMS processes

SWISS QUANTUM

I. Preliminary configuration

- Create a group on KEMS
- Set the KEMS server IP address on each QNC

II. QNC configuration

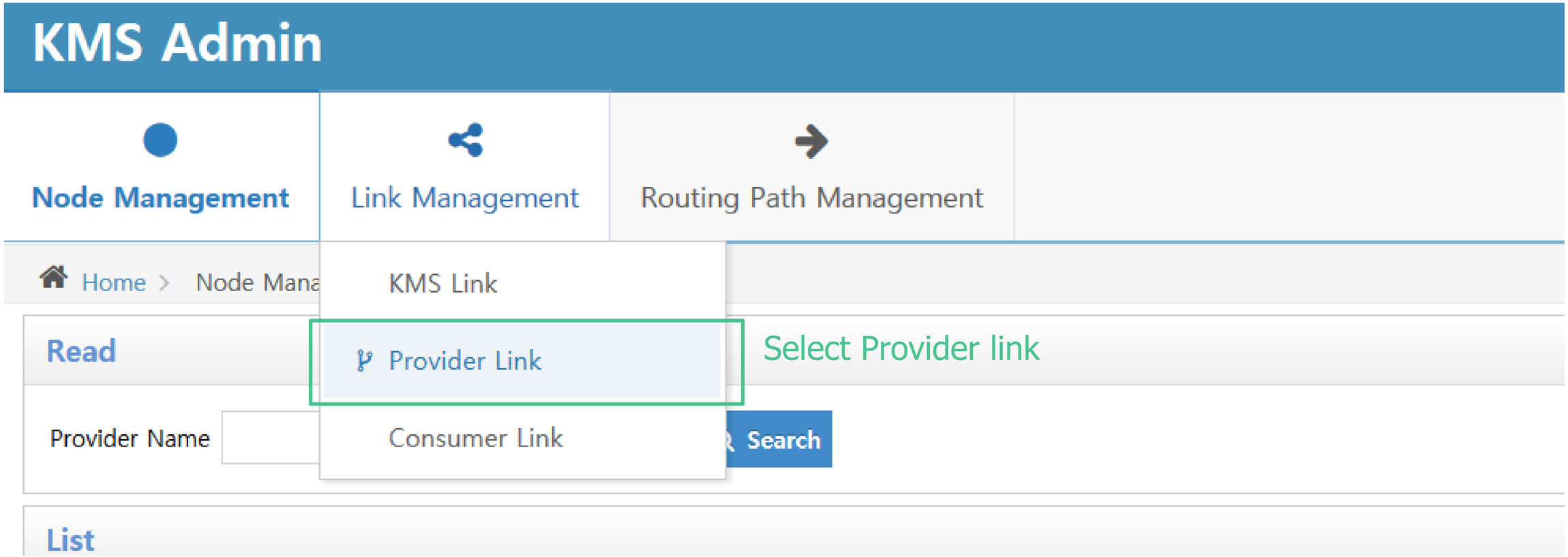
- Set the QNC name on each QNC and write down the UID
- Create each QNC on the KEMS
- Check if the KEMS server is connected to each QNC
- Create the QNC link between the both QNCs

III. QKD configuration

- Set each QKD on the KEMS
- Create the QKD link between both QKDs

IDQ

KEMS configuration



KEMS configuration

SWISS QUANTUM

Detail information

Insert the session name (link name of QKD)

Basic information

Session name

Operational Mode: Master

Source Register

Name | KMS Address

Vendor | KMS VID

Destination Register

Name | KMS Address

Vendor | KMS VID

Select Alice

Select Bob

Q key Store Info

Max Key Count: 100

Default Key Size: 256 bit

Crt Store Rate: 20 %

Max Key Count Per Request: 1

Key Request Interval: 250 mSec.

Store Rate: 40 %

Store Rate: 30 %

Source lookup

Name | Search

Name	Vendor
QKDA	Cerberis3

KMS VID	KMS Address
V2000000100001	QKDA

Select | Cancel

Destination lookup

Name | Search

Name	Vendor
QKDB	Cerberis3

KMS VID	KMS Address
V2000000200001	QKDB

Select | Cancel

IDQ

KEMS configuration

- Max Key Count: Number of keys to store (recommended: 20)
- Default Key Size: Use default value unless you're quite sure
- Max Key Count Per Request: Number of keys for request. Use default value
- Max Key Size: Max key size provider can dispense can request, Use default value
- Min Key Size: Use default value
- Key Request Interval: Key request interval in millisecond, Use default value
- Min Store Rate: Minor Rate Threshold to retain, Use default value
- Maj Store Rate: Major Rate Threshold to retain, Use default value
- Crt Store Rate: Critical Rate Threshold to retain, Use default value

KEMS configuration

SWISS QUANTUM

Detail information

Basic information

Session name: PROVM_PROVS

Operational Mode: Master

Source

Name	QKDA	KMS Address	QKDA
Vendor	Cerberis3	KMS VID	V2000000100001

Destination

Name	QKDB	KMS Address	QKDB
Vendor	Cerberis3	KMS VID	V2000000200001

Q key Store Info

Max Key Count	20	Max Key Count Per Request	1	Max Key Size	256 bit	Min Store Rate	40 %
Default Key Size	128 bit	Key Request Interval	250 mSec.	Min Key Size	128 bit	Maj Store Rate	30 %
Crt Store Rate	20 %						

New Save delete

ID Quantique PROPRIETARY

IDQ

KEMS processes

SWISS QUANTUM

IV. Encryptors configuration

- Create the encryptors on KEMS
- Create the link between the encryptors
- Add certificates on encryptors

V. Trusted path

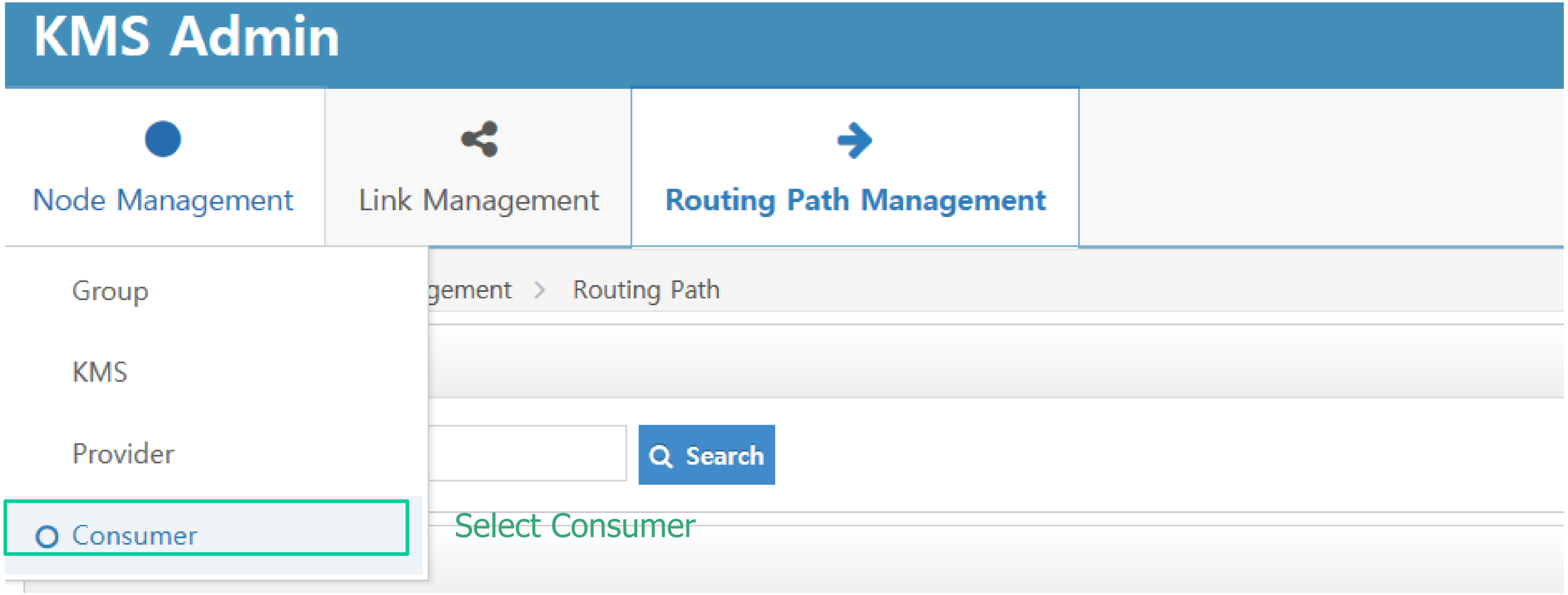
- Create the trusted path
- Deploy the trusted path

VI. Checking

- Check if the system is properly working and exchanging keys

IDQ

KEMS configuration



KEMS configuration

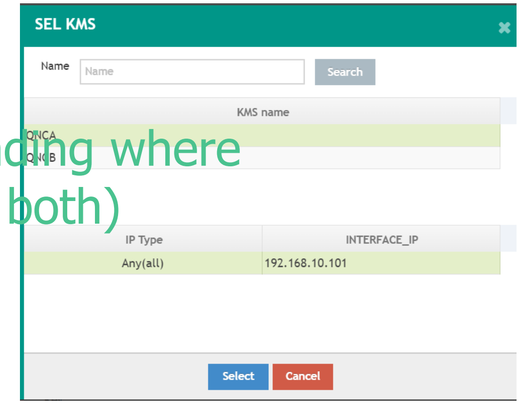
SWISS QUANTUM

Detail information

Associated KMS Info

Select QNC Mgmt or Frontpane IP (Depending where consumer is connected to… or 0.0.0.0 for both)

KMS Info	QNCA	Any(all)	192.168.10.101	Select

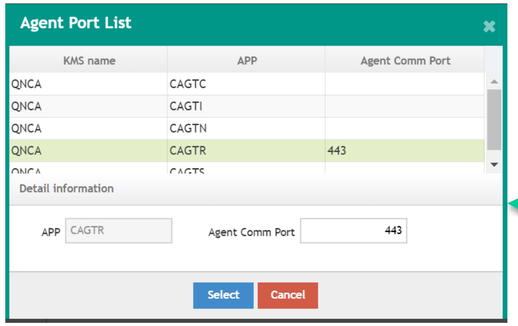


Configure 2x:
One for each Encryptor

Consumer Info

Agent Location	Inside KMS
Consumer Name	encalice
Protocol	ETSI REST 014
Agent Comm Port	Select

Select inside KMS
Alphanumeric letters for consumer identifier



Select CAGTR For ETSI
Select Port 443 For ETSI

Select CAGTI For IDQ3P

Select CAGTC For Cisco
Select Port 9999 for Cisco

- For consumer (encryptor) with **ETSI** interface choose '**ETSI REST 014**'
- For consumer (encryptor) with **IDQ3P** interface choose '**IDQ KEYSP**'
- For consumer (encryptor) with Cisco SKS interface choose "Cisco REST SKS"

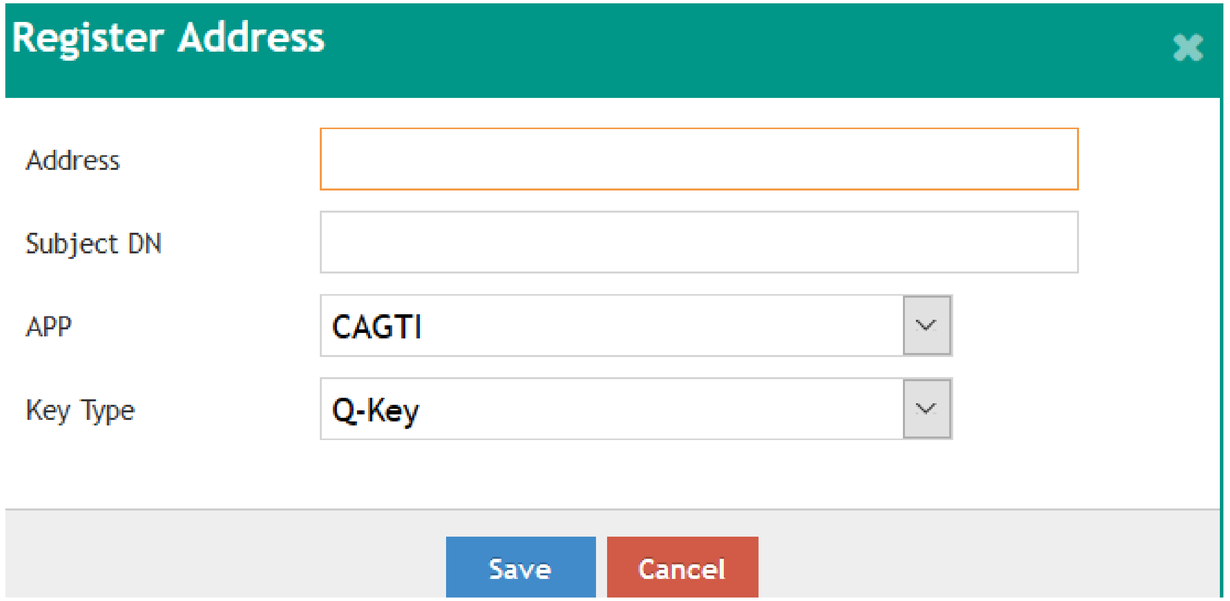
IDQ

KEMS configuration

To configure encryptors or ETSI client:

(or any name)

- **Address:** name of the encryptors that corresponds to the SAE Name per ETSI Standard
- **Subject DN:** Put the subject DN which appears in the client certificate (**No Spaces between separator**)
- **Agent Comm port:**
  - CAGTR (for ETSI Encryptor)
  - CAGTI (for IQD3P Encryptor)
  - CAGTC (for Cisco SKS)



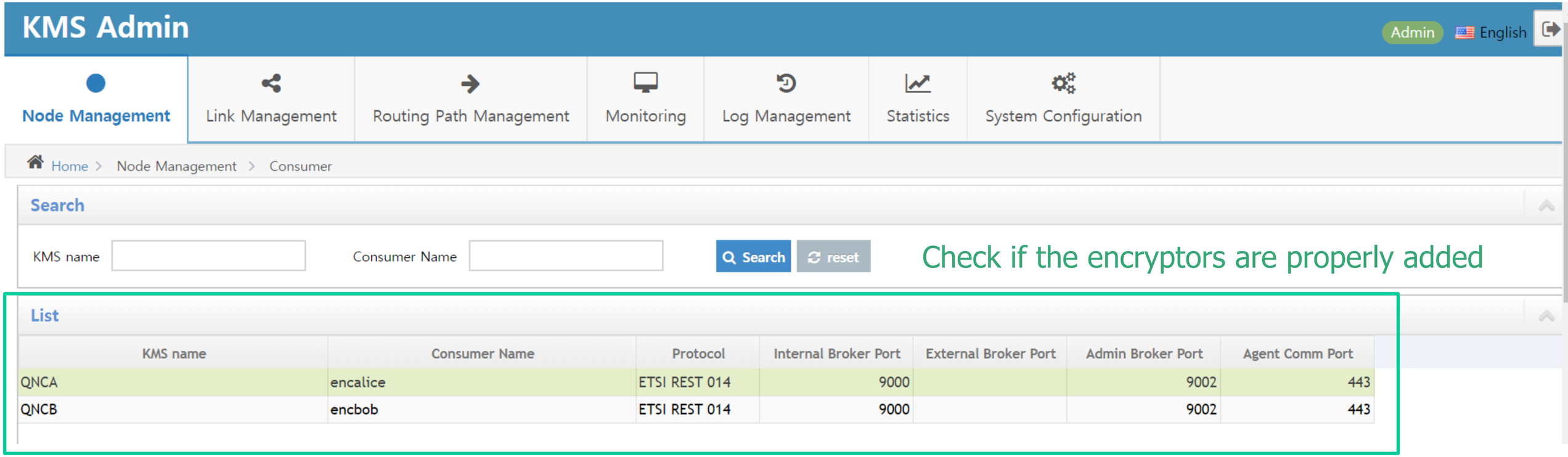
For more information

Go to slide 235+

Configure 2x: One for each Encryptor

When an encryptor does an ETSI key request (as master), it should use the slave name as SAE, as written in the ETSI standard. When the slave encryptor requests its “key by id”, the SAE is now the master one (we cross the SAE names).

KEMS configuration



KMS name	Consumer Name	Protocol	Internal Broker Port	External Broker Port	Admin Broker Port	Agent Comm Port
QNCA	encalice	ETSI REST 014	9000		9002	443
QNCB	encbob	ETSI REST 014	9000		9002	443

Check if the encryptors are properly added

KEMS processes

IV. Encryptors configuration

- Create the encryptors on KEMS
- Create the link between the encryptors
- Add certificates on encryptors

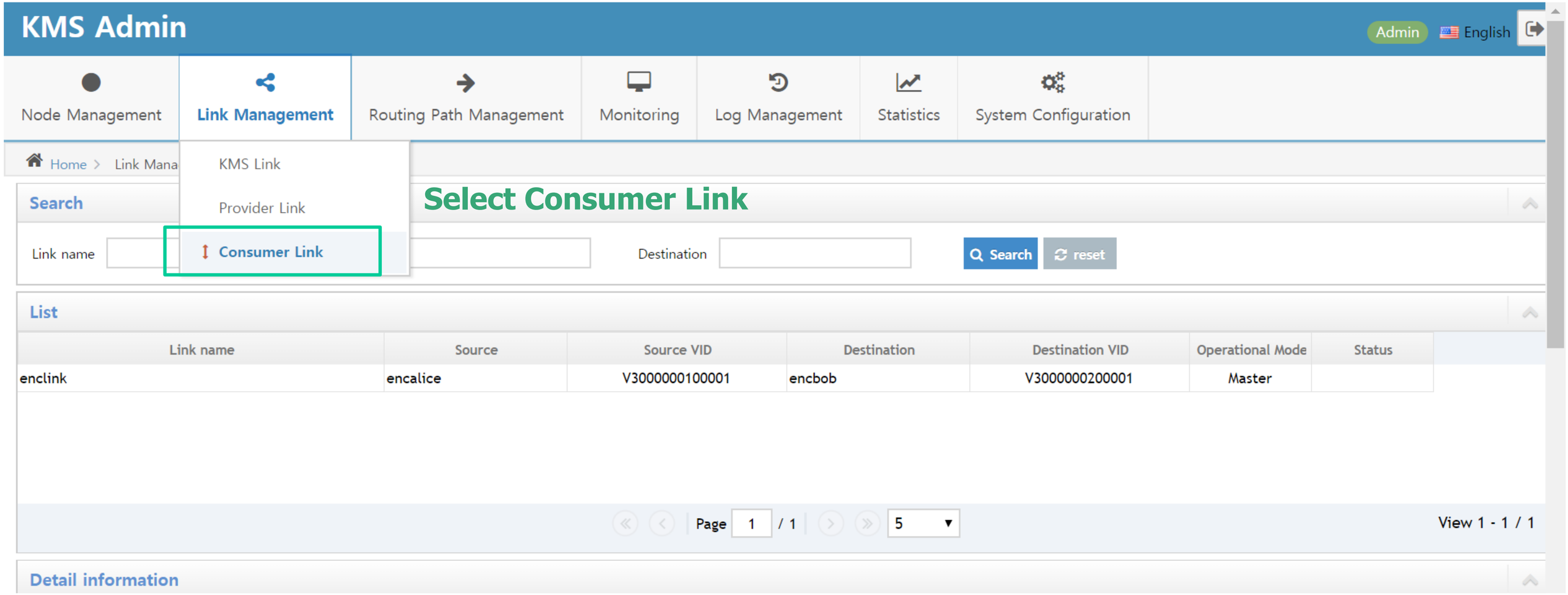
V. Trusted path

- Create the trusted path
- Deploy the trusted path

VI. Checking

- Check if the system is properly working and exchanging keys

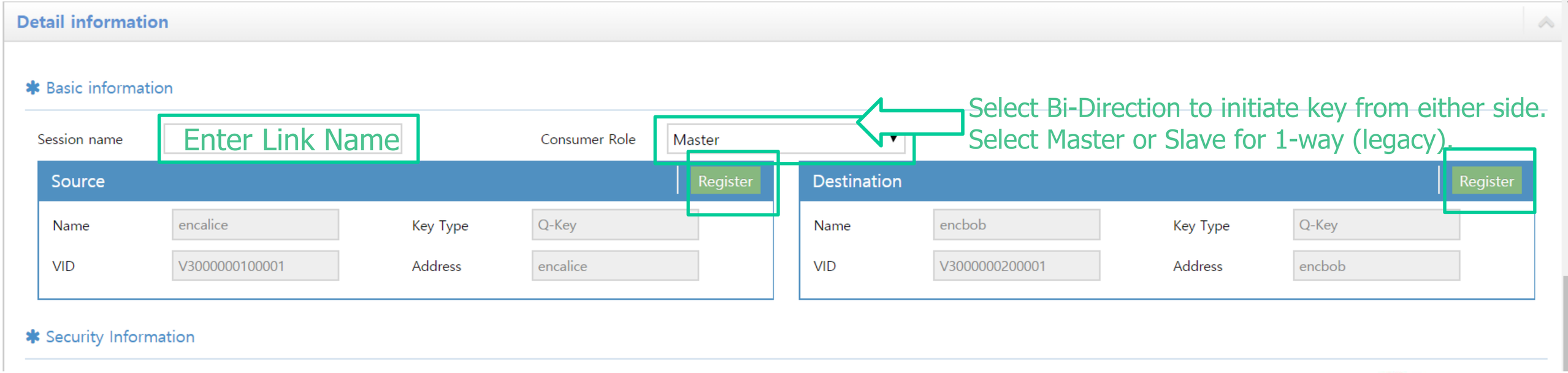
KEMS configuration



KEMS configuration

Select ALICE

Select BOB



KEMS configuration

- Use default values for Consumer Key Store Information.
- The default PSK is: 8BE4935F32866EE40FDB3F29ADF14BCC35B48C92ED89F08D1E0145B45BCC147368C97382CDB7D169B956EE4AA39E8BF7

✱ Security Information

PSK 8BE4935F32866EE40FDB3F29ADF14E

Put the Default PSK
(Today we only accept 127-byte Hex value)
(Note limitations today with Cisco in separate config doc)
Cisco PSK can be configured using QNET (ascii) instead of KEMS

✱ Consumer Key Store Information

Max Key Count	100	Max Key Count Per Request	1	Max Key Size	256 bit	Min Store Rate	40 %
Default Key Size	256 bit	Key Request Interval	250 msec	Min Key Size	128 bit	Maj Store Rate	30 %
Crt Store Rate	20 %						

New Save

KEMS processes

IV. Encryptors configuration

- Create the encryptors on KEMS
- Create the link between the encryptors
- Add certificates on encryptors

V. Trusted path

- Create the trusted path
- Deploy the trusted path

VI. Checking

- Check if the system is properly working and exchanging keys

Add Initial Authentication Key

To create and synchronize the initial authentication key, plug a USB stick into the 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 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)

ID Quantique PROPRIETARY

KEMS configuration

Configure 2x: (Alice & Bob)

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

- **Add the certificate authority in QNC**:
  - Plug the USB key with the certificates and keys into the QNC USB port
  - Execute the command in QNC CLI: ***encryptor --add-client-ca*** then choose which client root CA to use
- **Load the server Certificate and keys (QNC)**:
  - Plug the USB key with the certificates and keys into the QNC USB port
  - Execute the command in QNC CLI: ***encryptor --load-key*** then copy server keys and server certificates on the system

Note: Plug the USB into the Port N°1 of the QNC.

Note: One limitation of the system is that the client root CA must end with a blank line in order to be successfully parsed by the CLI command.

```
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 unwated CA files

```
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

KEMS configuration

Set the server certificates and keys:

- Execute the command QNC CLI: ***encryptor set-server-keys*** -> choose which server certificate and key to use to connect to the encryptor

Configure 2x: (Alice & Bob)

Note: All the files need to respect the X509 norm, the extension is not a problem (it can be .pem, .cert, .crt, .key,…) as long as the X509 format is followed.
File name must end in .pem

The consumers you created in KEMS 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

Use standalone command "encryptor" to verify configuration

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

KEMS processes

SWISS QUANTUM

IV. Encryptors configuration

- Create the encryptors on KEMS
- Create the link between the encryptors
- Add certificates on encryptors

V. Trusted path

- Create the trusted path
- Deploy the trusted path

VI. Checking

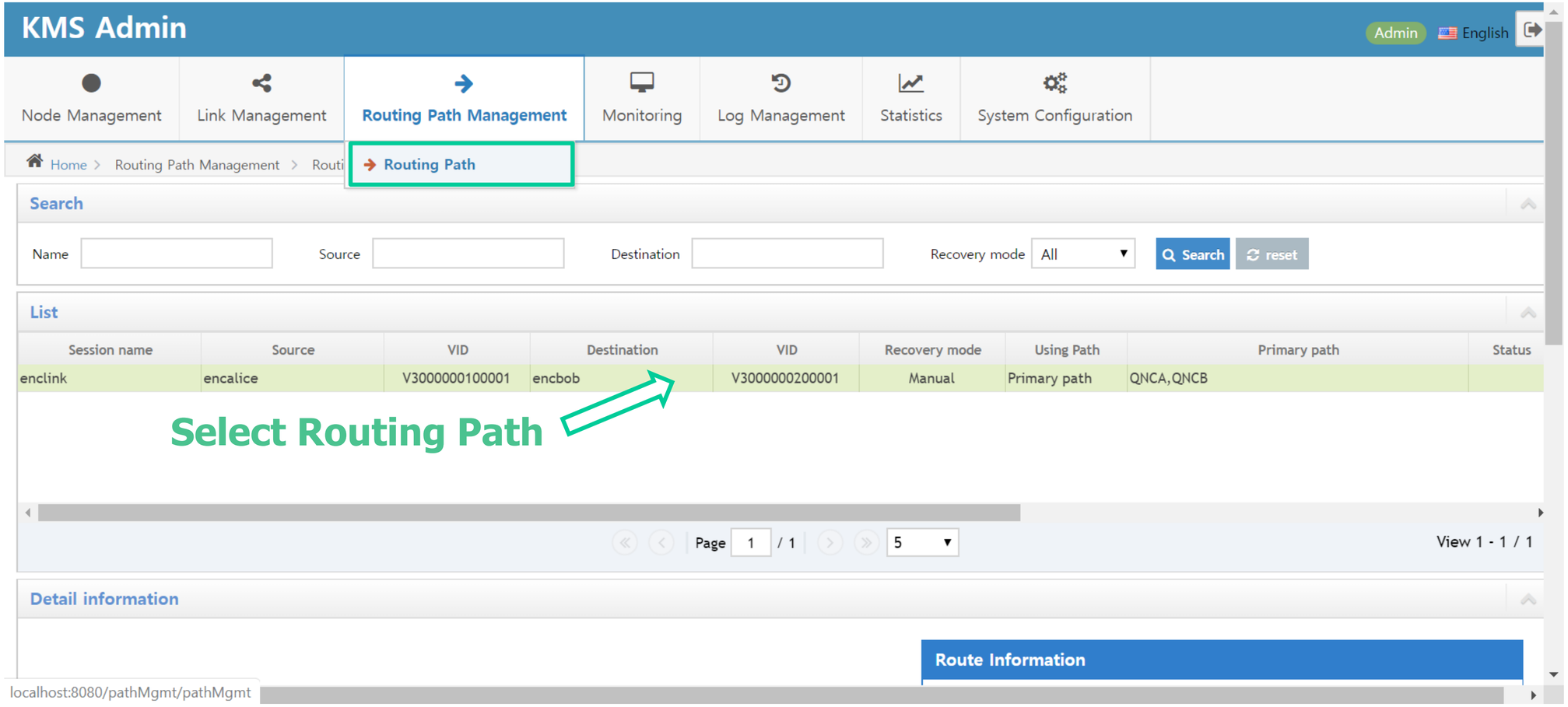
- Check if the system is properly working and exchanging keys

IDQ

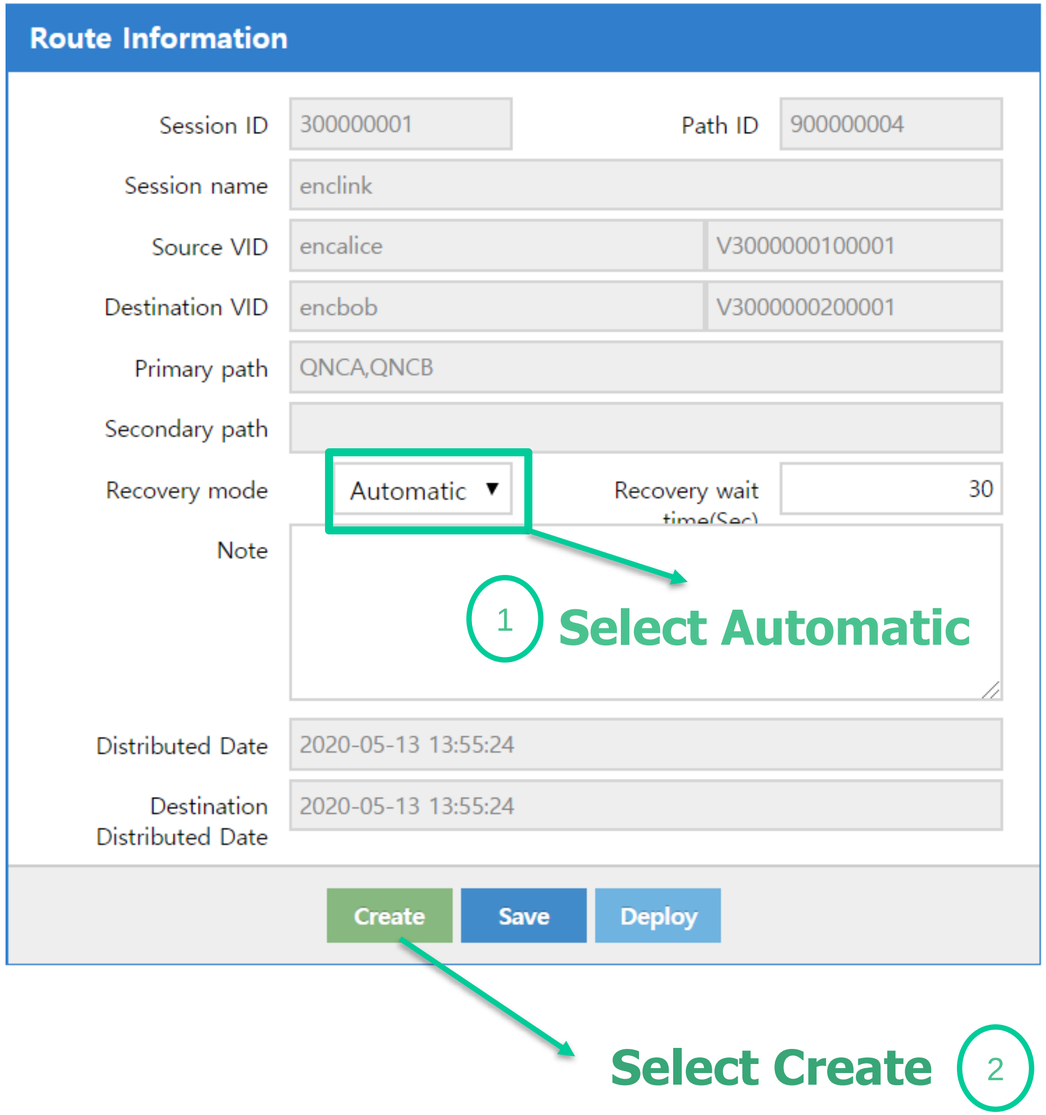
KEMS configuration

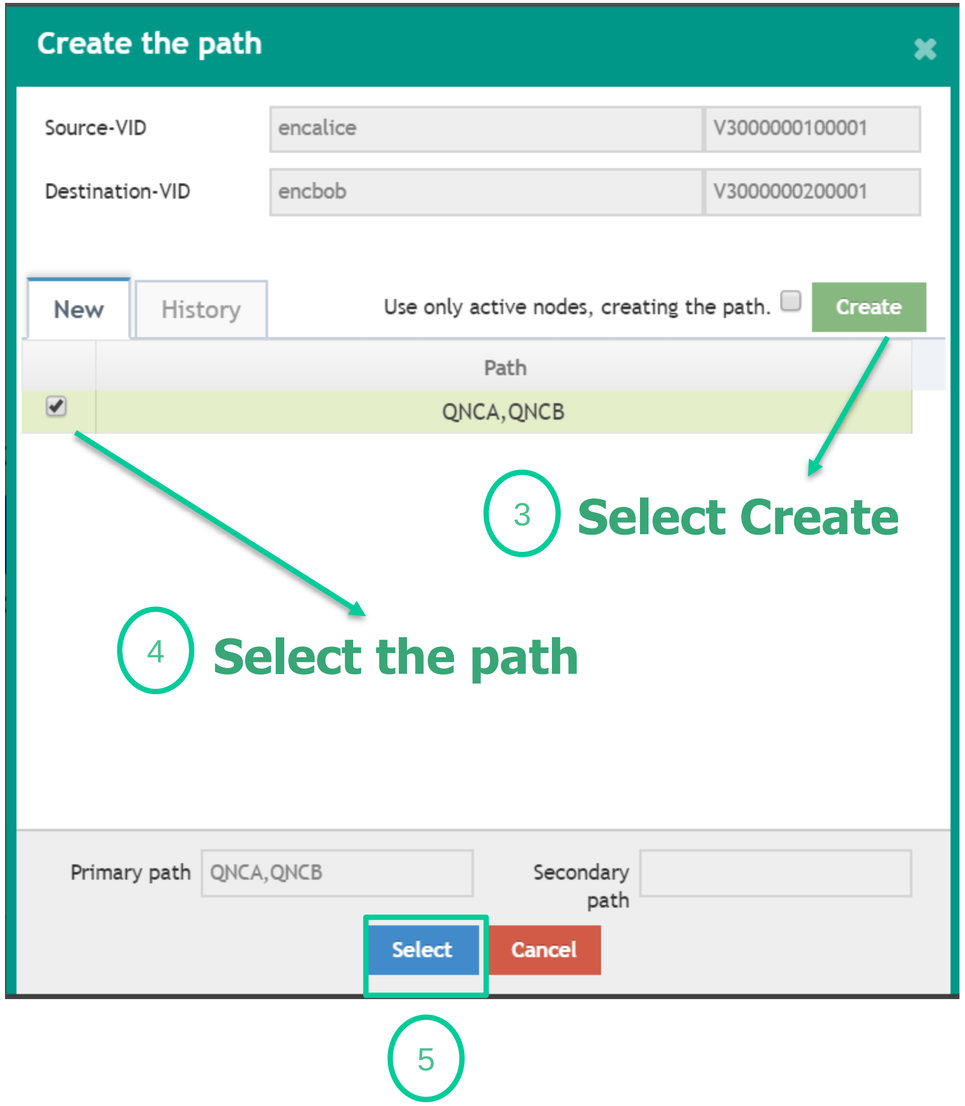
In the KEMS GUI:

- Go to: **Routing Path Management / Routing Path -> select consumer link**,
- Click Create and choose the KMS path, then click Save and Deploy



KEMS configuration





KEMS processes

SWISS QUANTUM

IV. Encryptors configuration

- Create the encryptors on KEMS
- Create the link between the encryptors
- Add certificates on encryptors

V. Trusted path

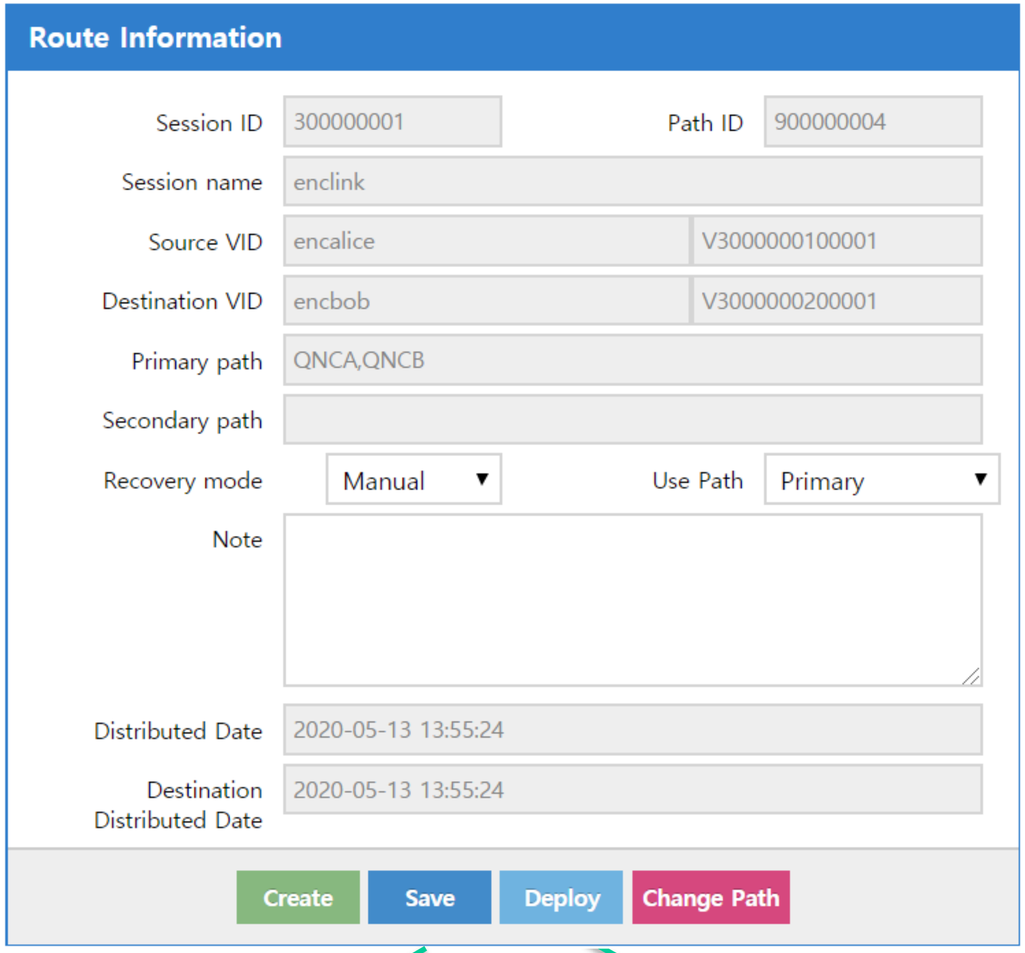
- Create the trusted path
- Deploy the trusted path

VI. Checking

- Check if the system is properly working and exchanging keys

IDQ

KEMS configuration



1 **Select Save**

Select Deploy 2

KEMS configuration

SWISS QUANTUM

Route Information

Session ID	300000001	Path ID	900000004
Session name	enclink		
Source VID	encalice	V3000000100001	
Destination VID	encbob	V3000000200001	
Primary path	QNCA,QNCB		
Secondary path			
Recovery mode	Manual	Use Path	Primary
Note			
Distributed Date	2020-05-13 13:55:24		
Destination Distributed Date	2020-05-13 13:55:24		

Create | Save | Deploy | Change Path

Check distributed date

IDQ

KEMS processes

SWISS QUANTUM

IV. Encryptors configuration

- Create the encryptors on KEMS
- Create the link between the encryptors
- Add certificates on encryptors

V. Trusted path

- Create the trusted path
- Deploy the trsuted path

VI. Checking

- Check if the system is properly working and exchanging keys

IDQ

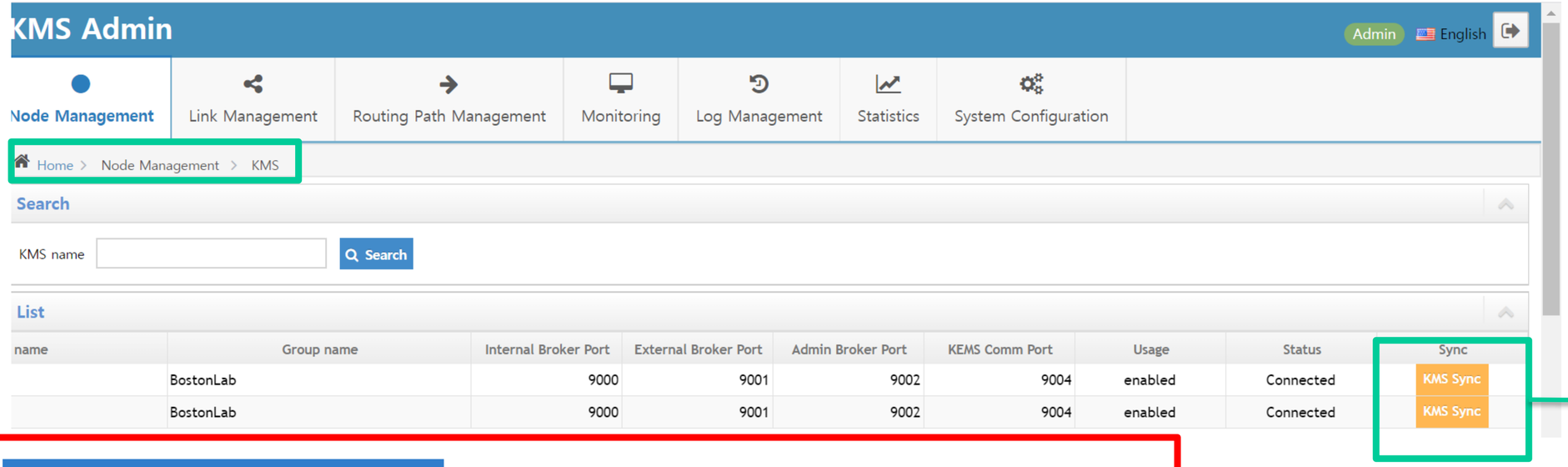
KEMS configuration

On QKD you should be able to see the set initial key

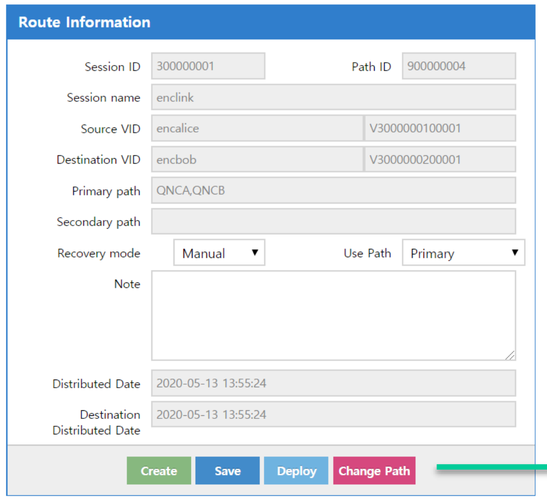
```
2020-02-19T16:36:55.374 DEBUG [(anonymous namespace)::FsmFrontend::NormalOperation_::ExecutingGeneralInitialization_*] Leaving
 ExecutingGeneralInitialization state.
2020-02-19T16:36:55.375 DEBUG [(anonymous namespace)::FsmFrontend::NormalOperation_::ExecutingSecurityInitialization*] Enterin
g ExecutingSecurityInitialization 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.
```

KEMS configuration

In the logs if you don't see « Received IDQ4P command 'SetInitialKey' » *you* may need to sync each KMS:



Press Sync for both KMS



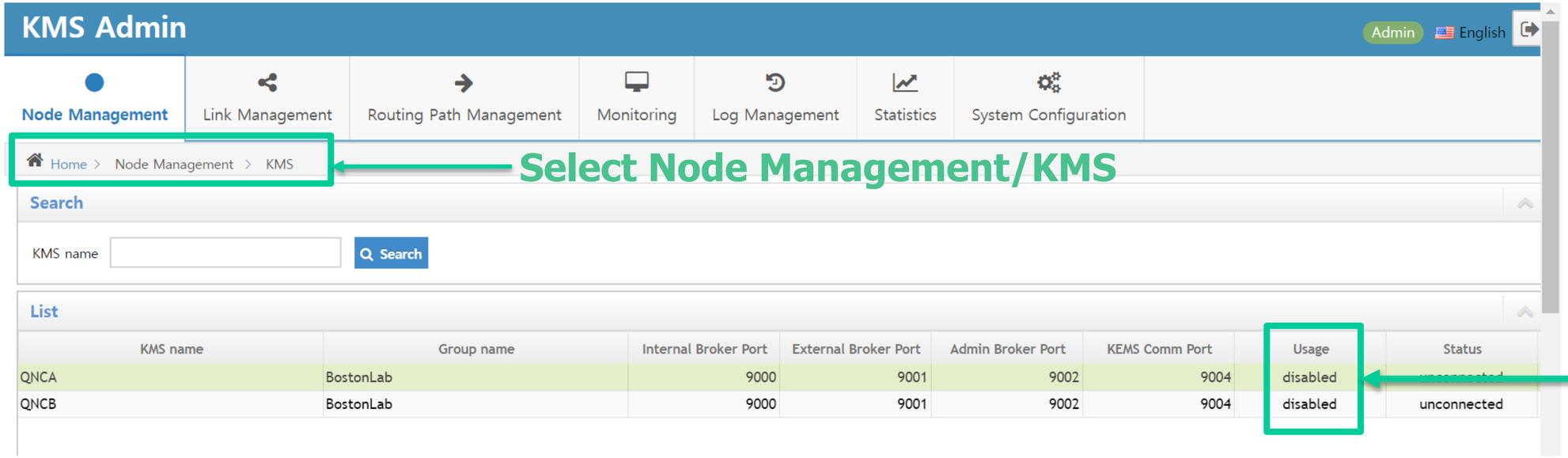
If KMS Sync does not work, redeploy the Routing Path and then try SYNC again.

Select Deploy

ID Quantique PROPRIETARY

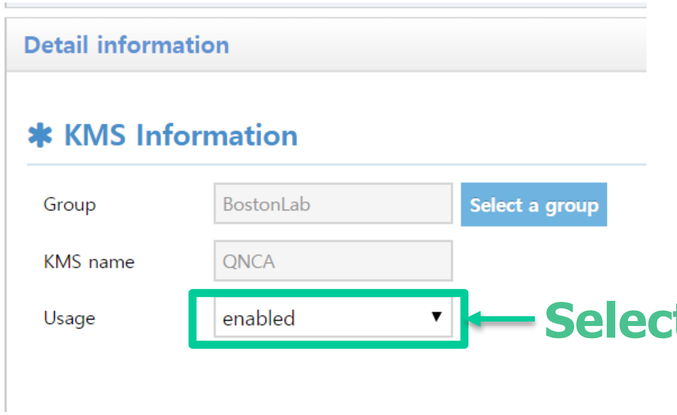
KEMS Configuration

Alternatively, the KMS status may have changed to "disabled" if the KMS was not able to communicate with the QNC after several attempts. Try to manually re-enable the KMS Usage:

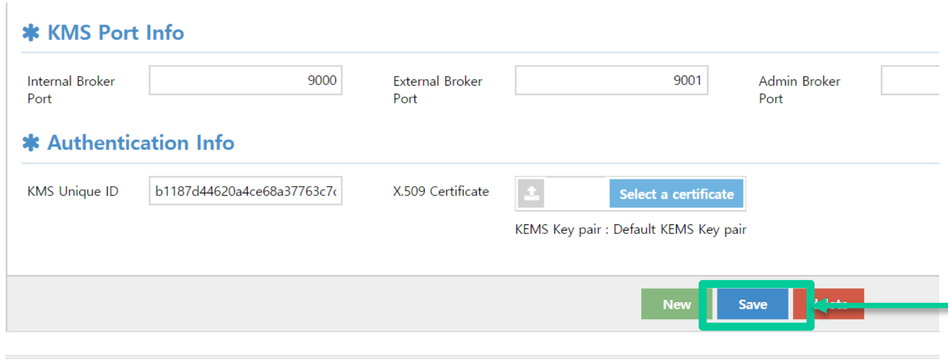


Select Node Management/KMS

Verify Here



Select "Enabled"



Click Save

ID Quantique PROPRIETARY

KEMS configuration

```
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

QNet Configuration API

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

- KEMS 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 KEMS configuration into XML/Json files for one node or a group of nodes
  - (Re) import XML/Json configuration into KEMS
  - Deploy configuration file directly into KMS nodes without the need of KEMS
  - Collect current configuration directly from KMS nodes

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

Sample QNet Commands (Script):

```
echo "-- Setting Login Credentials --"
qnet config qnetwebapi -url https://localhost:443 -user Admin -pwd Admin
printf "\n"

#FIRST DELETE THE GROUP IF IT ALREADY EXISTS
qnet delete group BostonLab
qnet create group BostonLab --desc BostonTestLab
qnet list group

qnet create node QNCA BostonLab -ip 192.168.10.101 --node-uid 12370e90d2114f42bdc669bf92b2f50c
#qnet update node QNCA -type Key -ip 192.168.20.20
qnet create node QNCB BostonLab -ip 192.168.10.106 --node-uid f7322a6ed6c544cda1fa845d1d292be4
#qnet update node QNCB -type Key -ip 192.168.20.30
qnet list node
qnet create node-link KMSLINK QNCA QNCB
qnet list node-link

qnet create provider QKDA QNCA Cerberis3 -ip 192.168.10.102
qnet create provider QKDB QNCB Cerberis3 -ip 192.168.10.107
qnet list provider
qnet create provider-link QKDLINK QKDA QKDB --max-key-count 20
qnet list provider-link

qnet create consumer centaurisA QNCA ETSI --subject-dn "C=AU,ST=Victoria,L=Melbourne,O=Org,OU=Security,CN=CN60100A0EAC" --address 192.168.10.120 --key-type QKEY
qnet create consumer centaurisB QNCB ETSI --subject-dn "C=AU,ST=Victoria,L=Melbourne,O=Org,OU=Security,CN=CN60100A0ED4" --address 192.168.10.130 --key-type QKEY
qnet list consumer
qnet create consumer-link CentaurisLink centaurisA centaurisB --mode BiDirectional
qnet list consumerlink

qnet create path centaurisA centaurisB -mode Automatic
qnet list path
qnet deploy group BostonLab
```

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
  loginspect  inspect 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>
```

loginspect **Inspect the system logs**
→ Provide view on all logs and archives

logtail **Monitor the system log**
→ Provide real-time view on the logs

logcp **Copy the system logs**
→ Copy logs on USB stick

QKD logs – loginspect command

```
admin> loginspect
Logs:
 0) communicatoralice.log
 1) communicatoralice.log.20201017-203210.gz
 2) communicatoralice.log.20201018-132313.gz
 3) communicatoralice.log.20201019-062003.gz

Please select a log between 0 (newest) and 3 (oldest)
```

last file's log

log archives

- loginspect works like the "less" command in linux

QKD logs – logtail command

Logtail's command

"live" logs

```
admin> logtail

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 --pattern to filter by type of information

- `

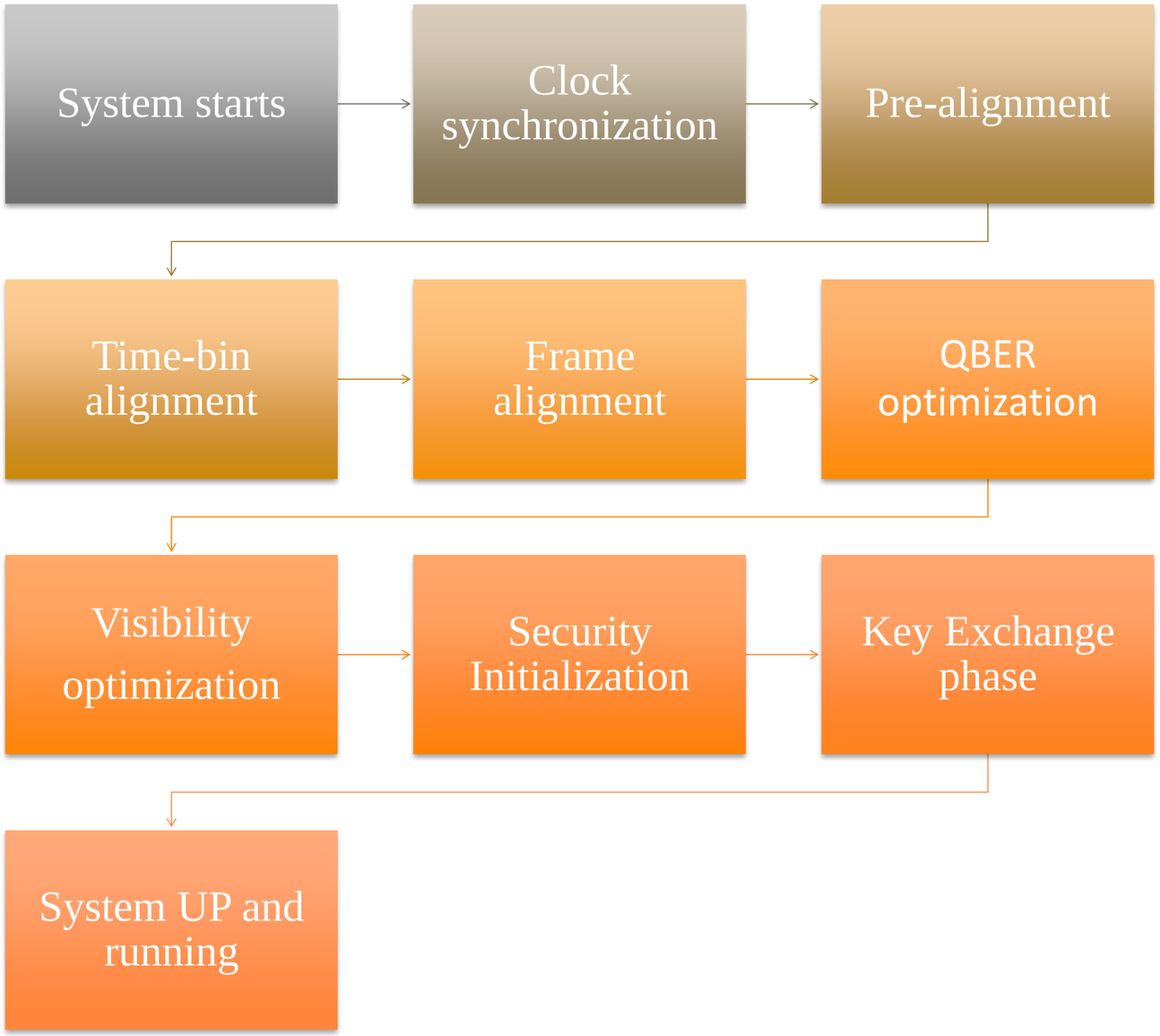
```
admin> logtail -h
Specify --help for a list of available options and commands.
Unrecognized option '-h'
admin> logtail --help
monitor the system logs

Usage: logtail [options] <log>

Arguments:
  log                       log to monitor
                            Allowed values are: qkd, 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**

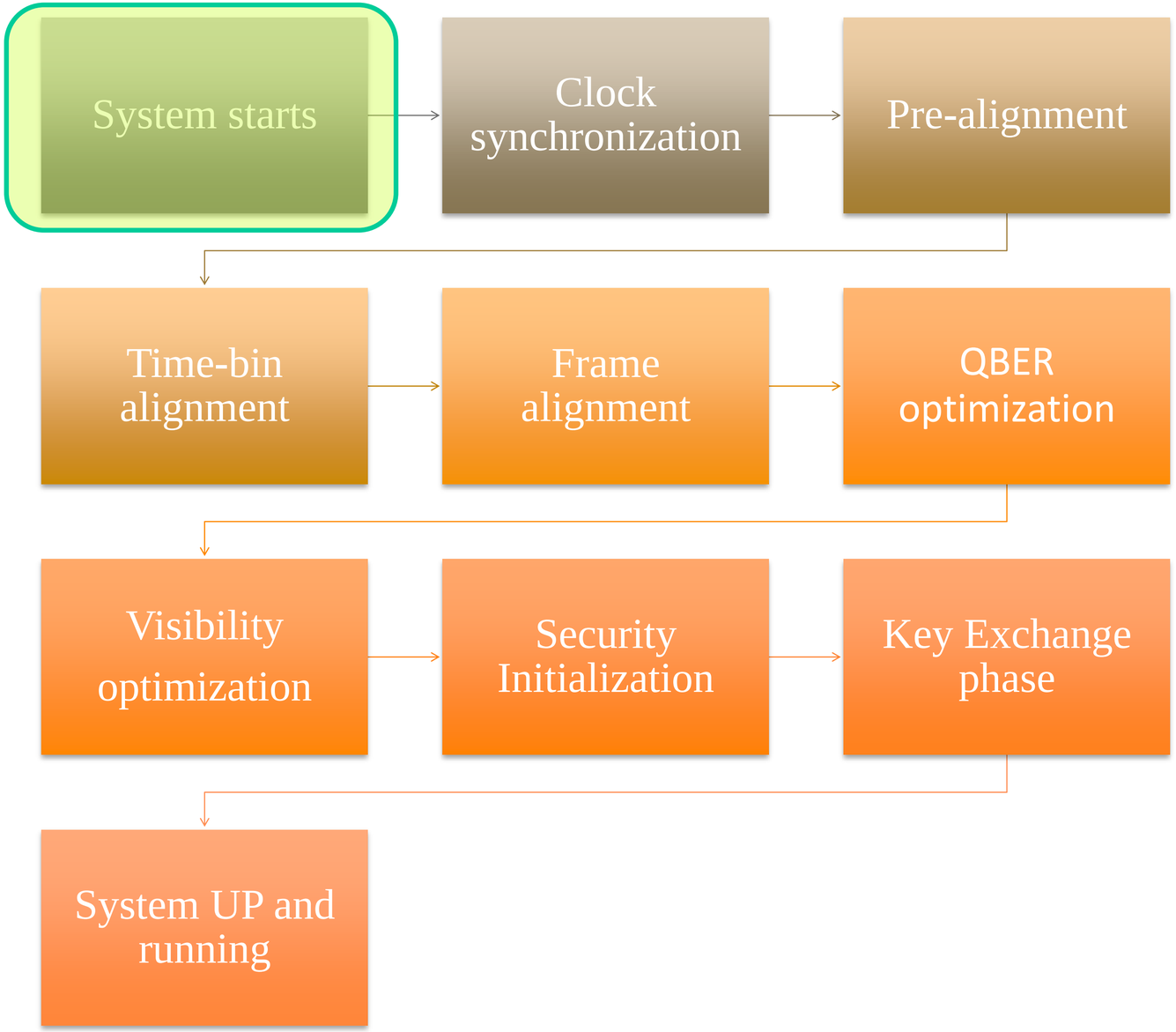
QKD logs – Understand the logs

- 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**

QKD logs – Understand the QKD processes



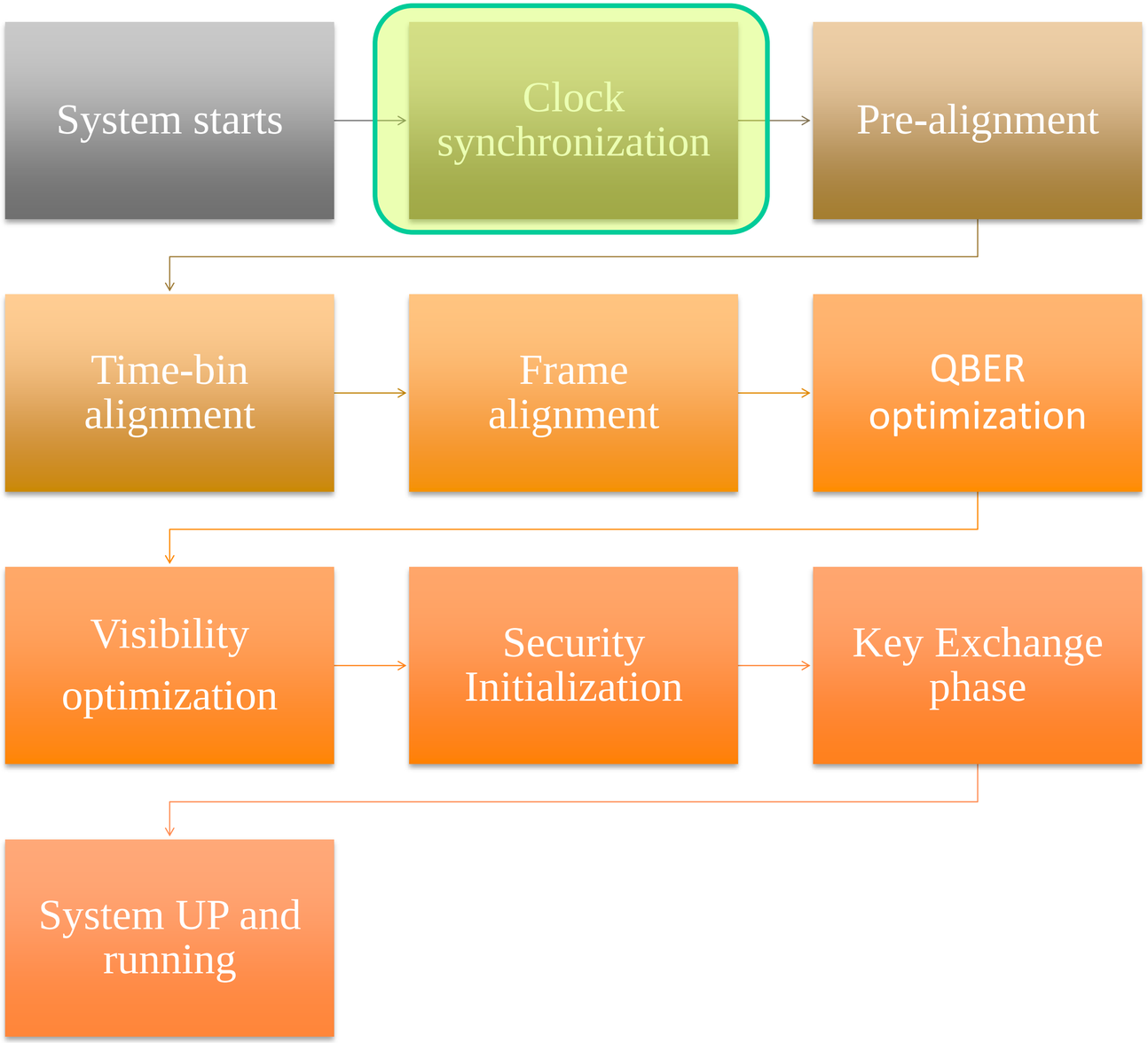
QKD logs – Understands the QKD processes



QKD logs – Understands the QKD processes

```
==========================================================
Communicator Alice
1.17.1.20200901083703 RELEASE
Copyright (C) 2015-2020 ID Quantique SA
==========================================================
IDQ4P   1.2.0
ZeroMQ  4.0.4
MsgPack 1.1.0
==========================================================

==========================================================
Communicator Bob
1.17.1.20200901083738 RELEASE
Copyright (C) 2015-2020 ID Quantique SA
==========================================================
IDQ4P   1.2.0
ZeroMQ  4.0.4
MsgPack 1.1.0
==========================================================
```

QKD logs – Understands the QKD processes



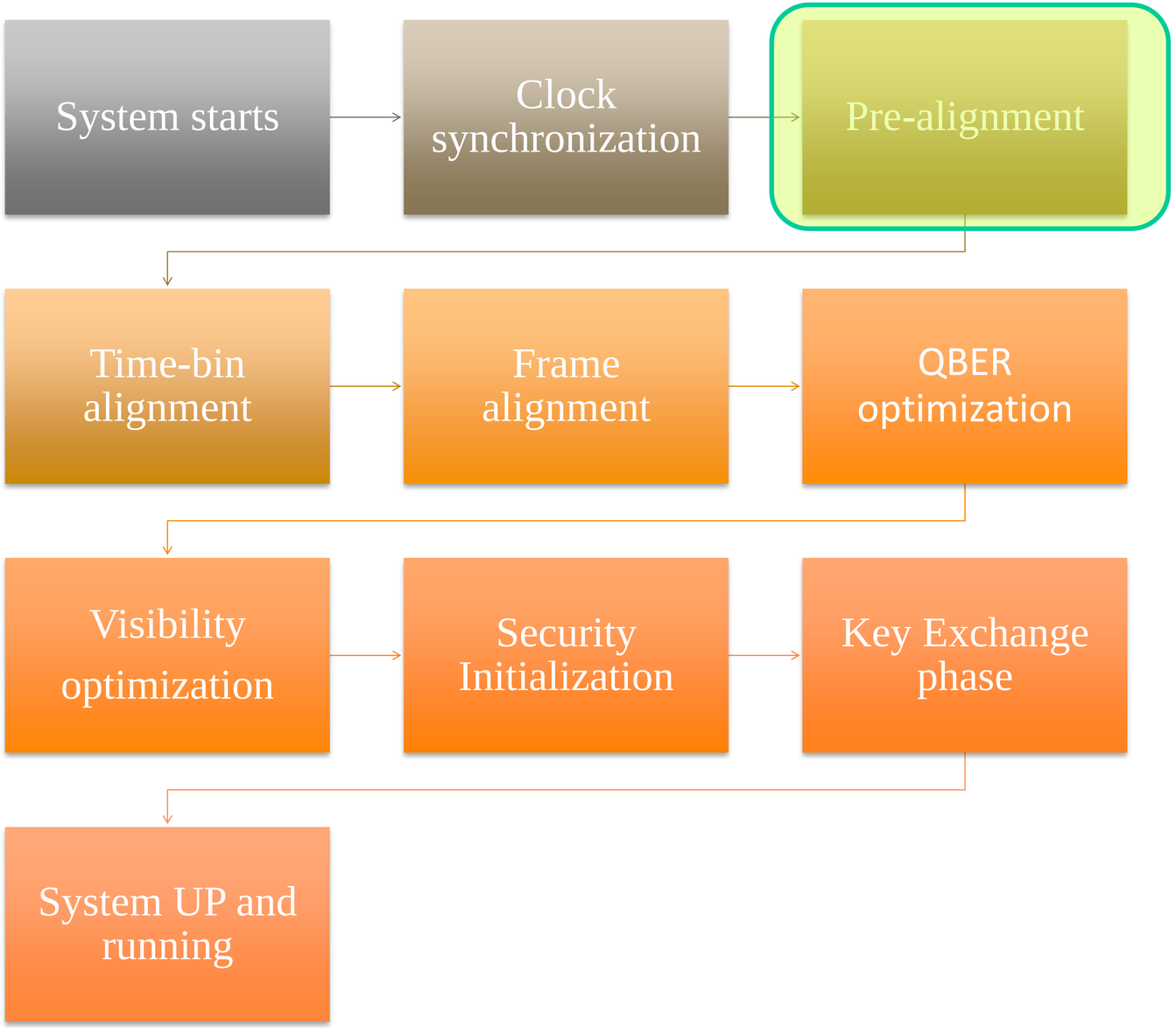
QKD logs – Understands the QKD processes

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

```
Local FPGA state:  SendCntSync
Remote FPGA state: SendFrame_Sync
Service channel restart count: 249
Clock not synchronized, resetting FPGA.
Checking PLL lock...
PLL locked.
…
Local FPGA state:  QCTest
Remote FPGA state: QCTest
Service channel restart count: 263
Service channel restart count check OK
Default configuration set.
```

Current state of the local FPGA.
Current state of the remote FPGA.
Counter FPGA restarts.
Indicates asynchronous FPGA clocks.
Indicates locking of the local PLL.
FPGA clocks synchronized.

QKD logs – Understands the QKD processes

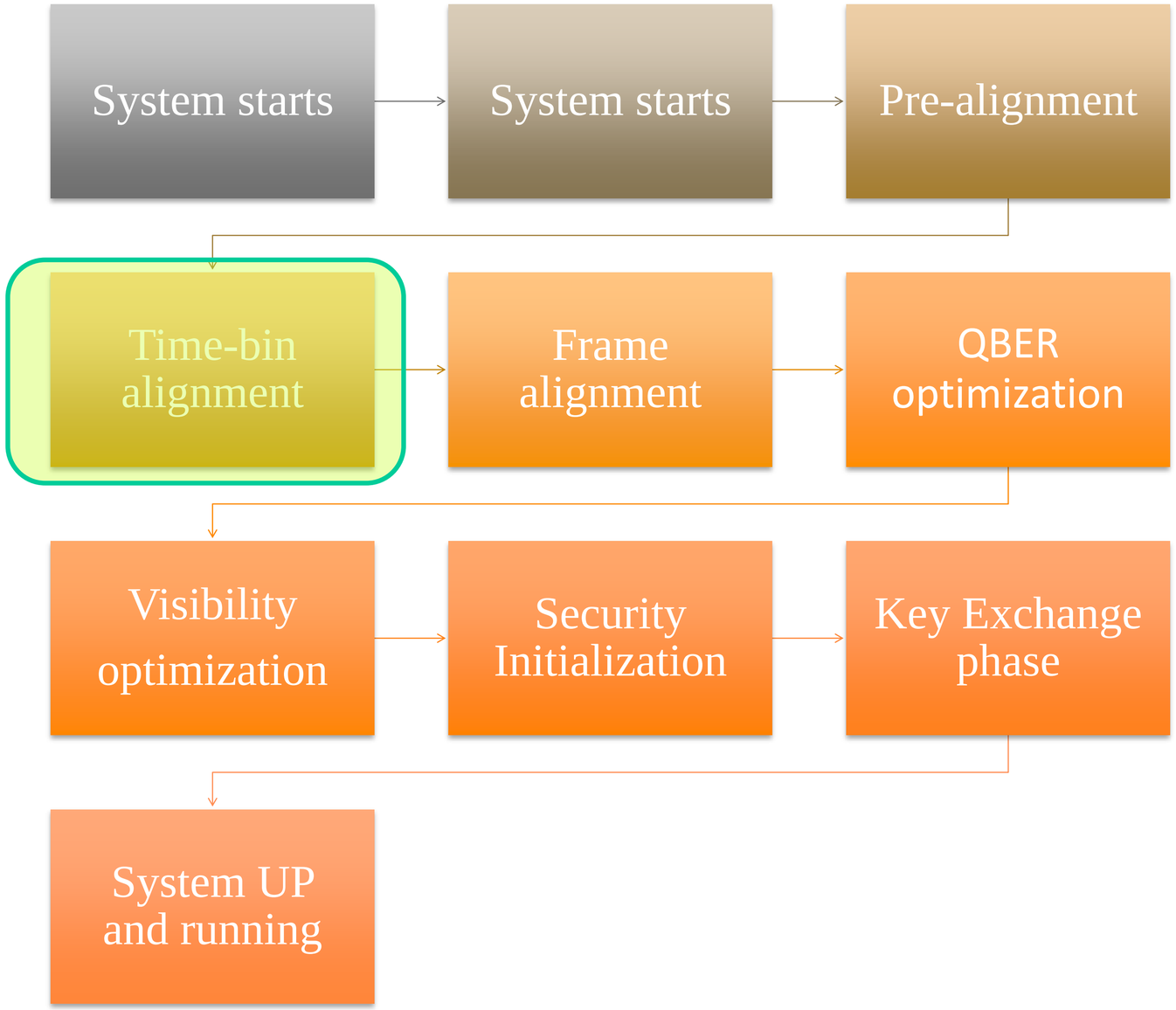


QKD logs – Understands the QKD processes

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

```
Optimizing IM power...
Pre-alignment pattern set to 0x88828
Optimizing IM power: bias voltage = -6.5 V, power = 0.0742653 A.
Optimizing IM power: bias voltage = -5.5 V, power = 0.0725988 A.
⋮                                                                  Coarse scan.
Optimizing IM power: bias voltage =  0.5 V, power = 0.0127260 A.
⋮
Optimizing IM power: bias voltage =  6.5 V, power = 0.0745986 A.
Optimizing IM power: highest coarse IM power = 0.0745986 A for bias voltage = 6.5 V.
Optimizing IM power: bias voltage = 5.5 V, power = 0.0712353 A.
⋮                                                                  Fine scan.
Optimizing IM power: bias voltage = 7.5 V, power = 0.0688113 A.
Optimizing IM power: highest fine IM power = 0.0745986 A for bias voltage = 6.2 V.
Optimizing IM power: found highest IM power for bias voltage = 6.2 V, power = 0.0745986 A.
IM power optimized.
```

QKD logs – Understands the QKD processes



QKD logs – Understands the QKD processes

- Wait until the DATA time-bin is aligned

```
Aligning DATA timebin...
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:
136 | 8709 | 1980 | 435 | 250 | 267 | 150 | 162 | 156 | 8720 | 1993 | 583 | 200 | 187 | 128 | 8522 | 2034 | 521 | 263 | 278
Aligning DATA timebin: delay = 2000, shift = 0, correlation = -0.230852.
⋮
Aligning DATA timebin: delay = 2000, shift = 19, correlation = 0.175895.
Aligning DATA timebin: temporary best alignment for delay = 2000, shift = 11, correlation = 0.977478.
⋮
Aligning DATA timebin: temporary best alignment for delay = 2275, shift = 10, correlation = 0.995198.
Aligning DATA timebin: found best alignment for delay = 2275, shift = 10, correlation = 0.995198.
DATA timebin aligned.
```

QKD logs – Understands the QKD processes

- Wait until the MONITOR time-bin is aligned

```
Aligning MONITOR timebin...
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
1413 | 11505 | 7753 | 11188 | 7307 | 3213 | 1507 | 11116 | 6585 | 11553 | 7570 | 3165 | 1557 | 1626 | 1280 | 12285 | 7146 | 11161 | 7031 | 2782
Aligning MONITOR timebin: delay = 2000, shift = 0, correlation = -0.509347.
⋮
Aligning MONITOR timebin: delay = 2000, shift = 19, correlation = 0.567688.
Aligning MONITOR pulse: temporary best alignment for delay = 2000, shift = 5, correlation = 0.829812.
⋮
Aligning MONITOR timebin: found best alignment for delay = 2605, shift = 3, correlation = 0.996137.
MONITOR timebin aligned...
```

QKD logs – Understands the QKD processes

SWISS QUANTUM

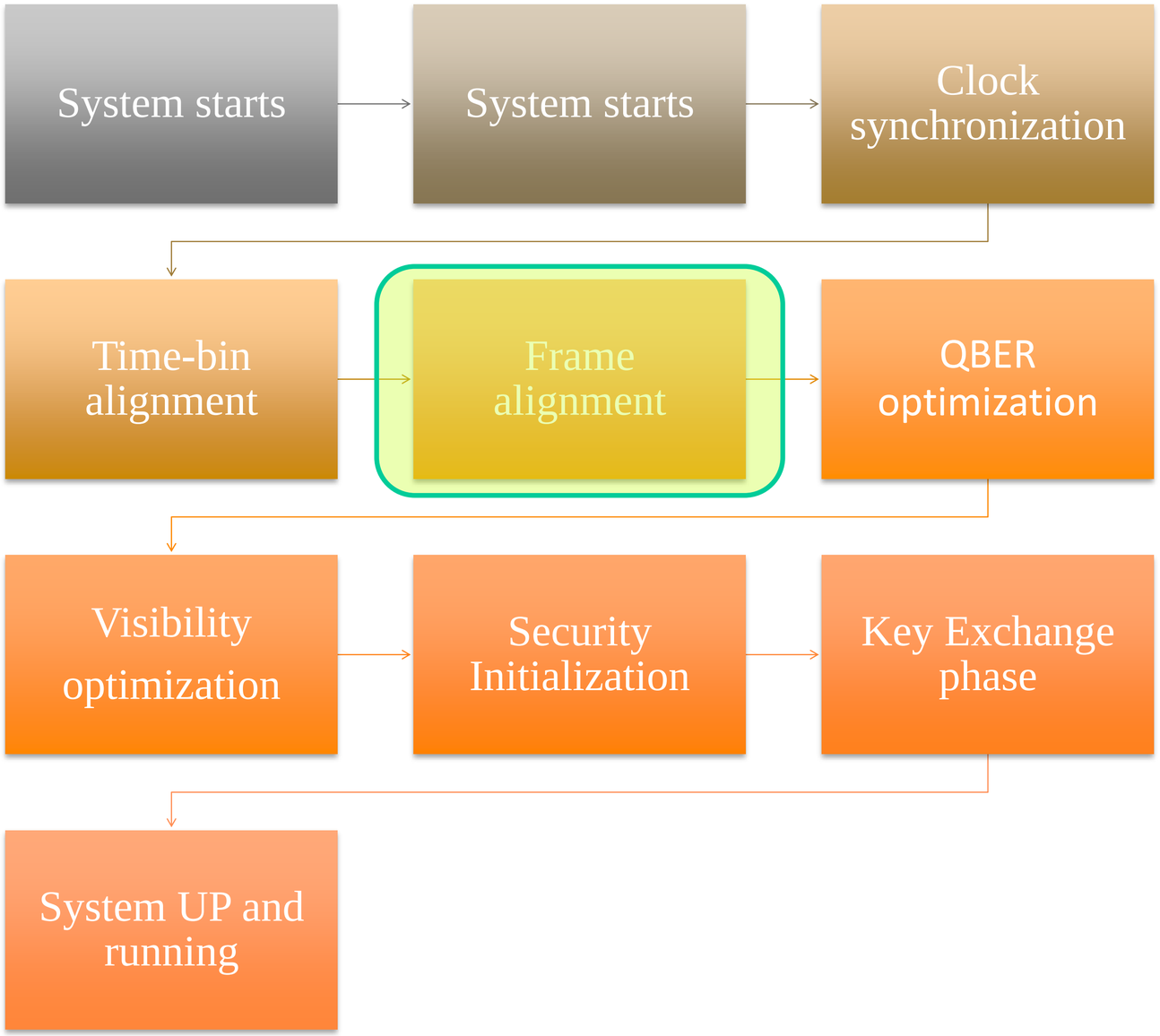
NOTE

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

IDQ

QKD logs – Understands the QKD processes

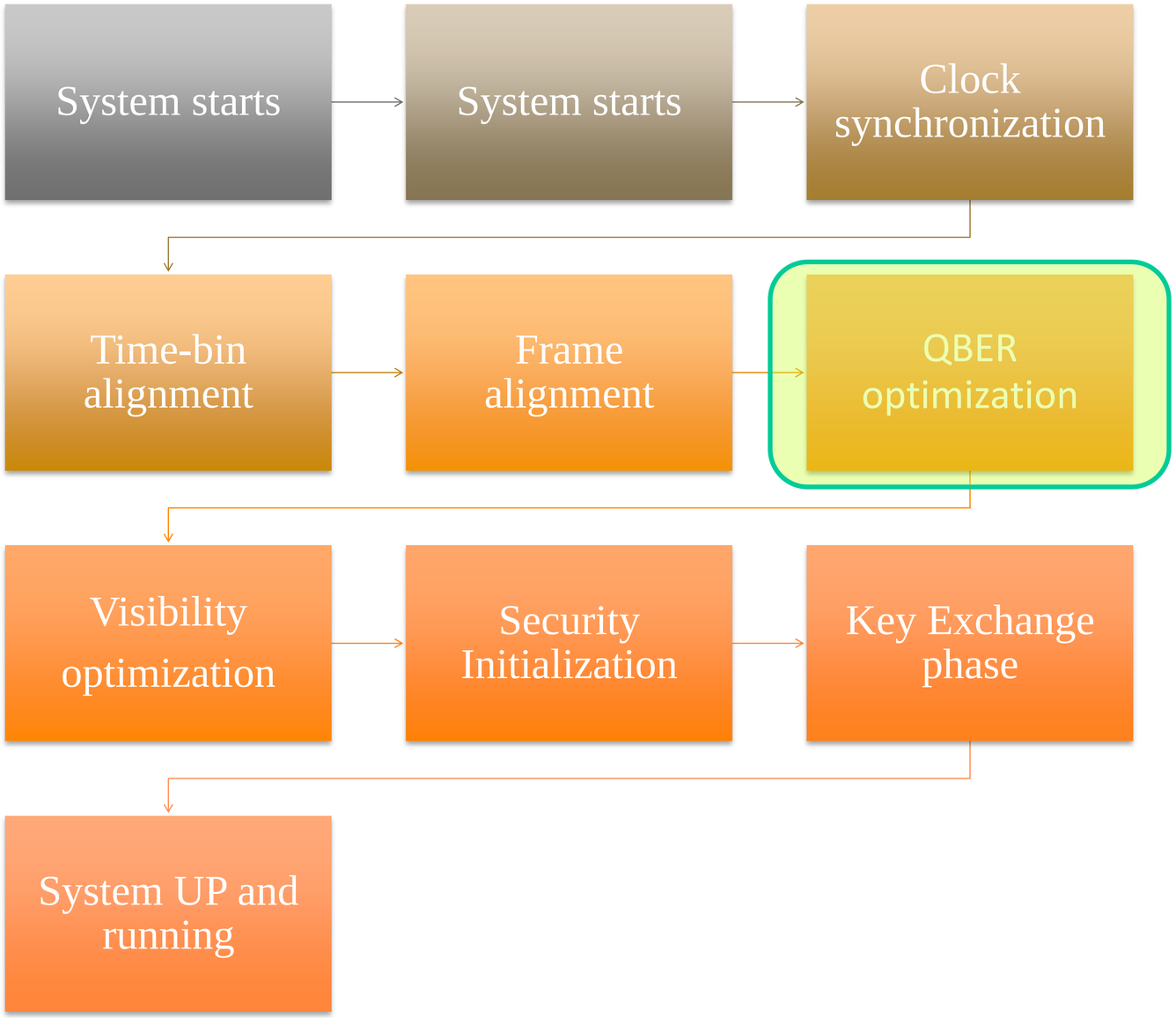


QKD logs – Understands the QKD processes

- Wait until the DATA & Monitor frame are aligned

```
   Aligning DATA frame...
1  Aligning DATA frame: expecting pattern 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0
   Aligning DATA frame: current timebins at delay = 18:
2  211 | 20 | 3 | 3 | 6 | 20 | 206 | 19 | 3 | 3 | 0 | 14 | 177 | 26 | 2 | 2 | 1 | 2 | 1 | 13
   Aligning DATA frame: delay = 18, correlation = -0.994158.
   ⋮
   Aligning DATA frame: current timebins at delay = 22:
   1 | 6 | 183 | 18 | 2 | 0 | 1 | 10 | 194 | 15 | 2 | 0 | 5 | 18 | 192 | 23 | 4 | 1 | 2 | 13
3  Aligning DATA frame: delay = 22, correlation = 0.999247.
4  Aligning DATA frame: found alignment at delay = 22.
   DATA frame aligned.
   Aligning MONITOR frame...
1  Aligning MONITOR frame: expecting pattern 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0
   Aligning MONITOR frame: current timebins at delay = 18:
2  604 | 89 | 569 | 59 | 48 | 89 | 642 | 98 | 586 | 66 | 56 | 66 | 557 | 93 | 599 | 70 | 44 | 34 | 31 | 81
   Aligning MONITOR frame: delay = 18, correlation = -0.995971.
   ⋮
   Aligning MONITOR frame: current timebins at delay = 22:
   49 | 89 | 582 | 79 | 584 | 71 | 45 | 105 | 591 | 98 | 561 | 71 | 38 | 81 | 606 | 93 | 563 | 62 | 49 | 73
3  Aligning MONITOR frame: delay = 22, correlation = 0.999533.
4  Aligning MONITOR frame: found alignment at delay = 22.
   MONITOR frame aligned.
```

QKD logs – Understands the QKD processes



QKD logs – Understands the QKD processes

- QBER should be around 0.01 and 0.03 max

- If QBER lower value is too high -> losses are too high, or problem with QC fiber connections

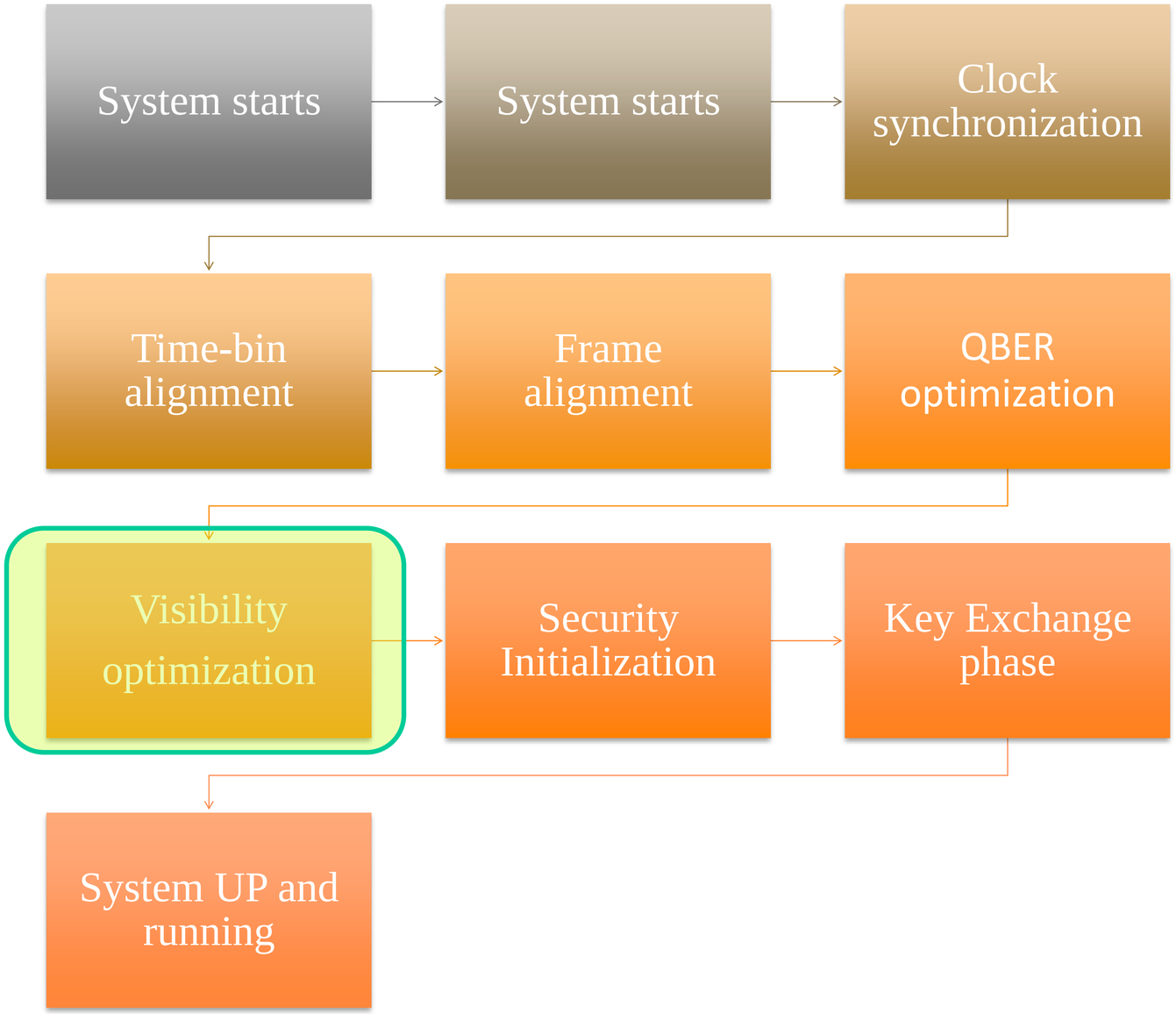
Alice

```
Minimizing QBER...                                   Starting the QBER Optimization step.
Waiting for remote FPGA Optics Optimization state 1 (current state: 0)...
⋮                                                    Waiting for Bob to start the QBER Optimization step.
Minimizing QBER: modulator bias = -6301 mV.
⋮                                                    Currently applied modulator bias voltages.
Minimizing QBER: modulator bias = -6051 mV.
QBER minimized.                                      Finalizing the QBER Optimization.
```

Bob

```
Minimizing QBER...                                   Starting the QBER Optimization step.
Remote FPGA Optics Optimization state: 0 (expected 1). Waiting for Alice.
Minimizing QBER: changing modulator bias value...
Minimizing QBER: modulator bias = -6301 mV.
Minimizing QBER: QBER = 0.0329128.                   Currently applied modulator bias voltages
⋮                                                    and corresponding measured QBER.
Minimizing QBER: modulator bias = -6051 mV.
Minimizing QBER: QBER = 0.0221098.
Minimizing QBER: found lowest QBER for …
QBER minimized.                                      Finalizing the QBER Optimization.
```

QKD logs – Understands the QKD processes



QKD logs – Understands the QKD processes

- Visibility is scanned
- Final visibility must be close to 1.000 (i.e. 0.90 is a good value)

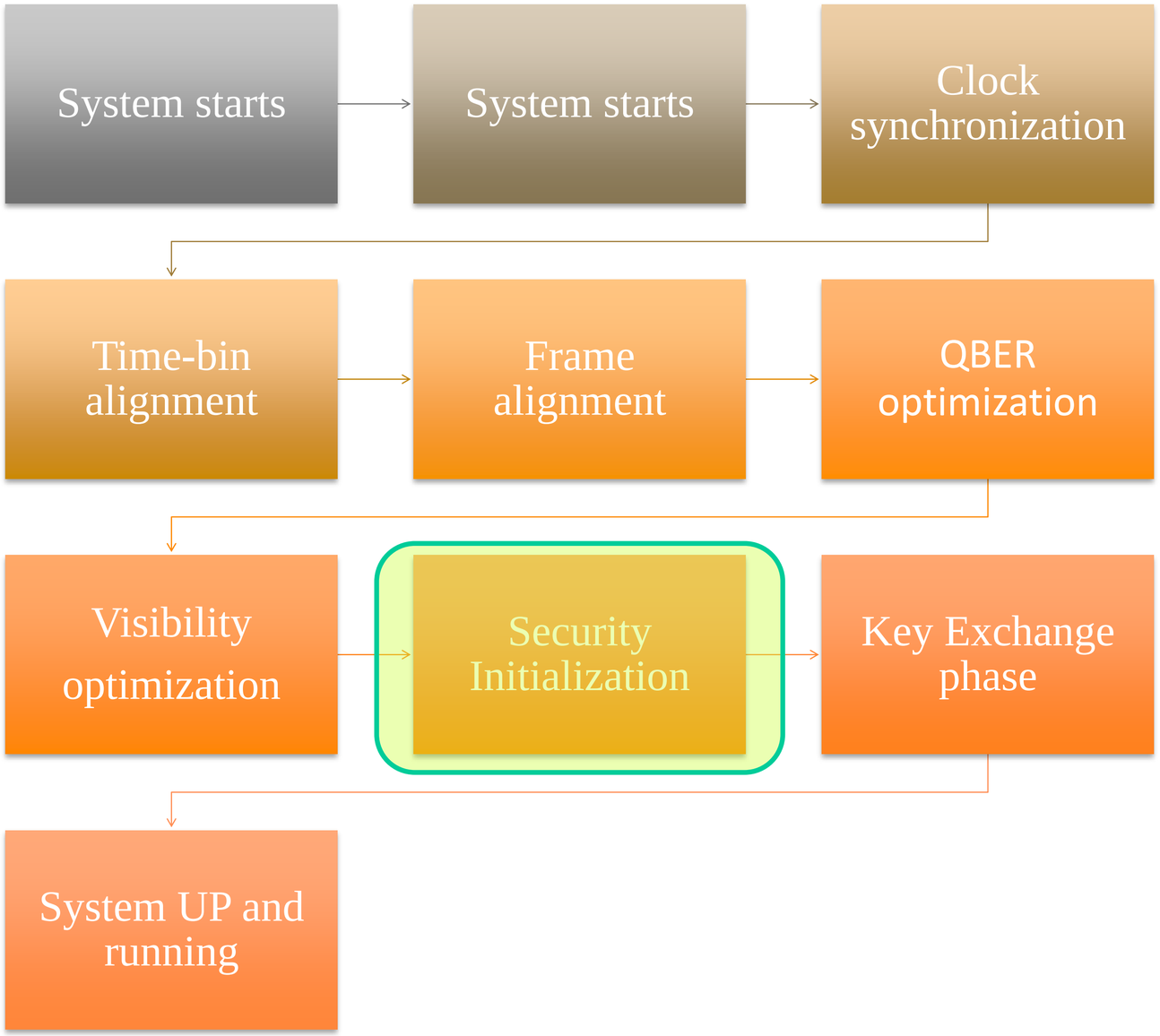
Alice

```
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.                Currently applied laser bias current and
Maximizing visibility: visibility = 0.467572.             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

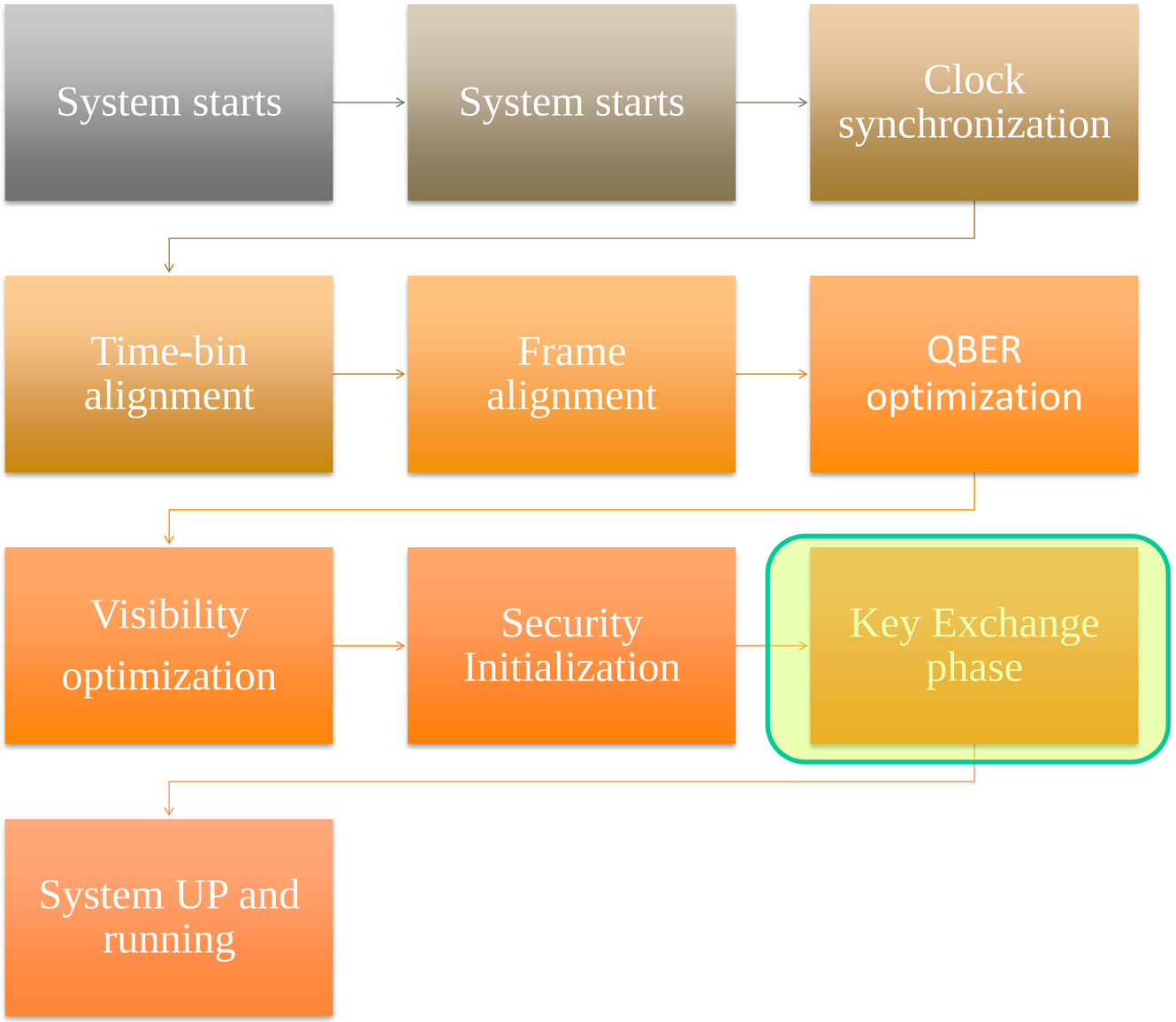


QKD logs – Understands the QKD processes

- The KEMS will be involved for this step
- If the initial key is not provided, you need to check if the KEMS server is running.
- If needed you could re-create – re-deploy the path and re-sync KMS

`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

- KEMS is involved.
- If KEMS is not reachable or configured, **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.
...
```

CERBERIS 3

SNMP monitoring

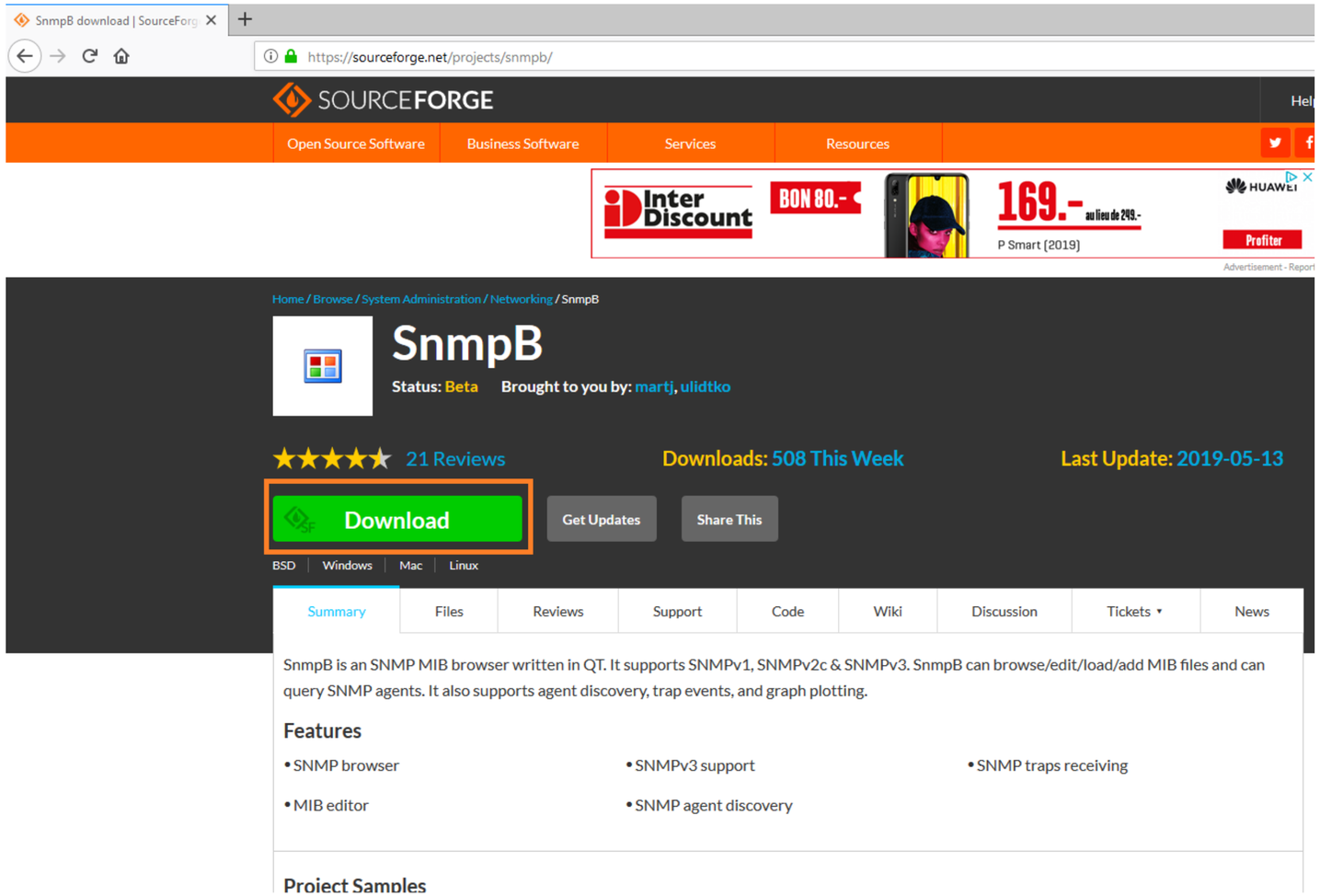
Management and monitoring

- QKD and QNC parameters and realtime values can be checked using:
    - SNMPv3 walk
    - Syslogs
    - Logs
- We recommend to use SNMPv3 walk to get the values.
- The recommended tool to use snmpwalk is SNMPB (open source)

SNMPB configuration

1. Download and Install SNMPB

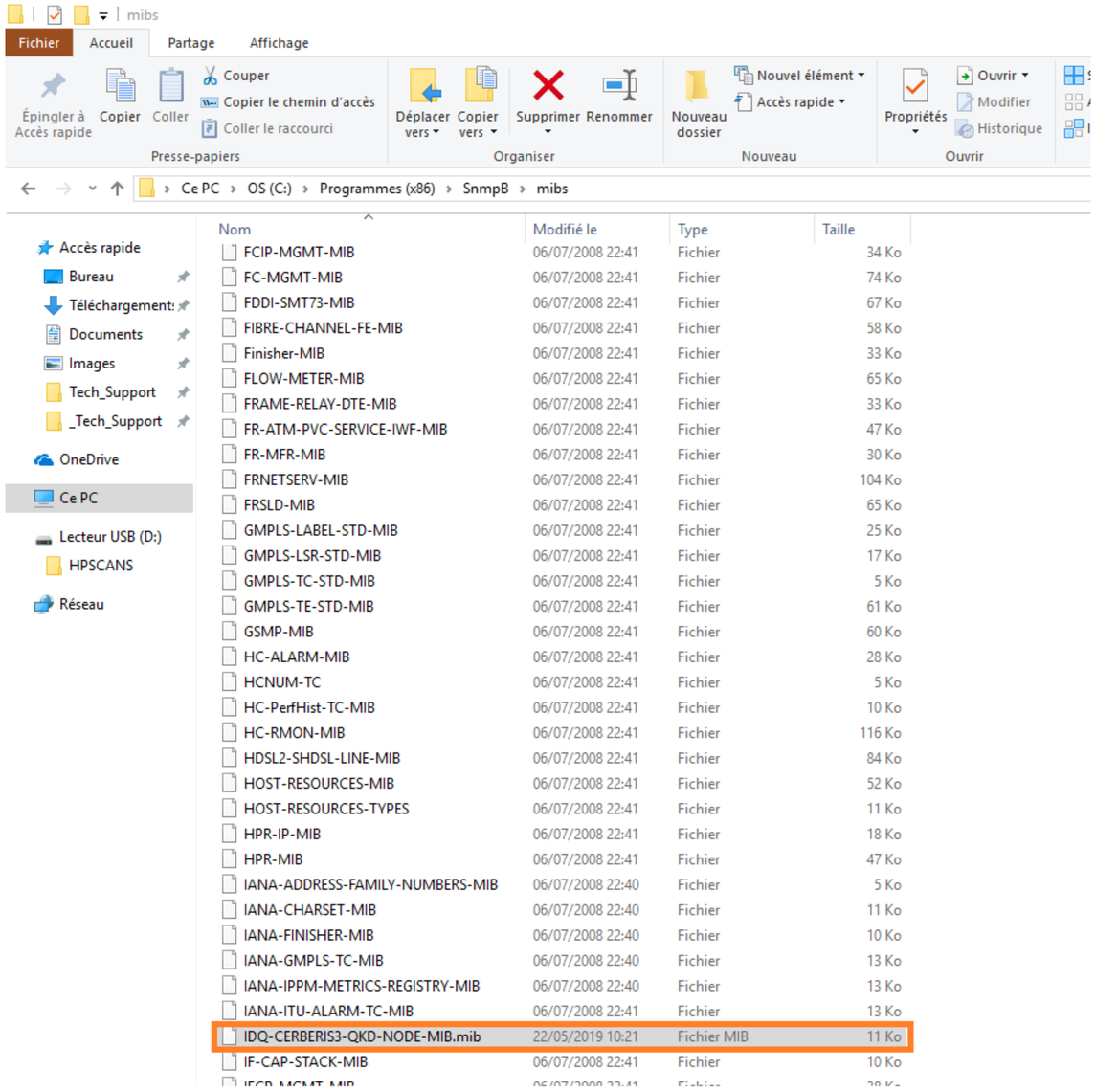
- Go to https://sourceforge.net/projects/snmpb/
- Download the file
- Install the tool



SNMPB configuration

2. Copy MIB file to root directory

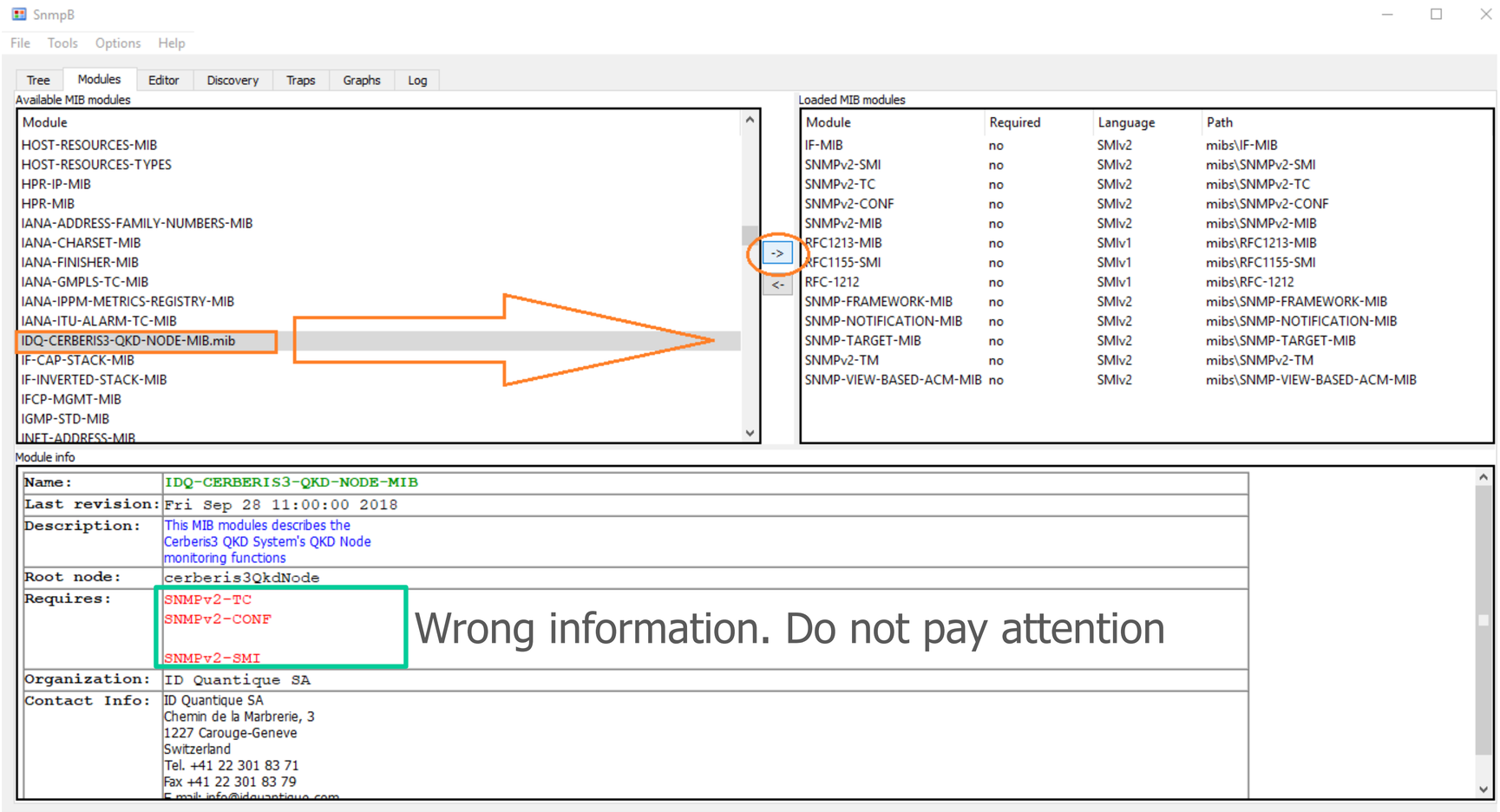
- Get the MIB files named:
  1. IDQ-CERBERIS3-QKD-NODE-MIB.mib
  2. VT-SH.mib
- Copy files to to C:\Program Files (x86)\SnmpB\mibs



SNMPB configuration

3. Configure MIBs on SNMPB

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



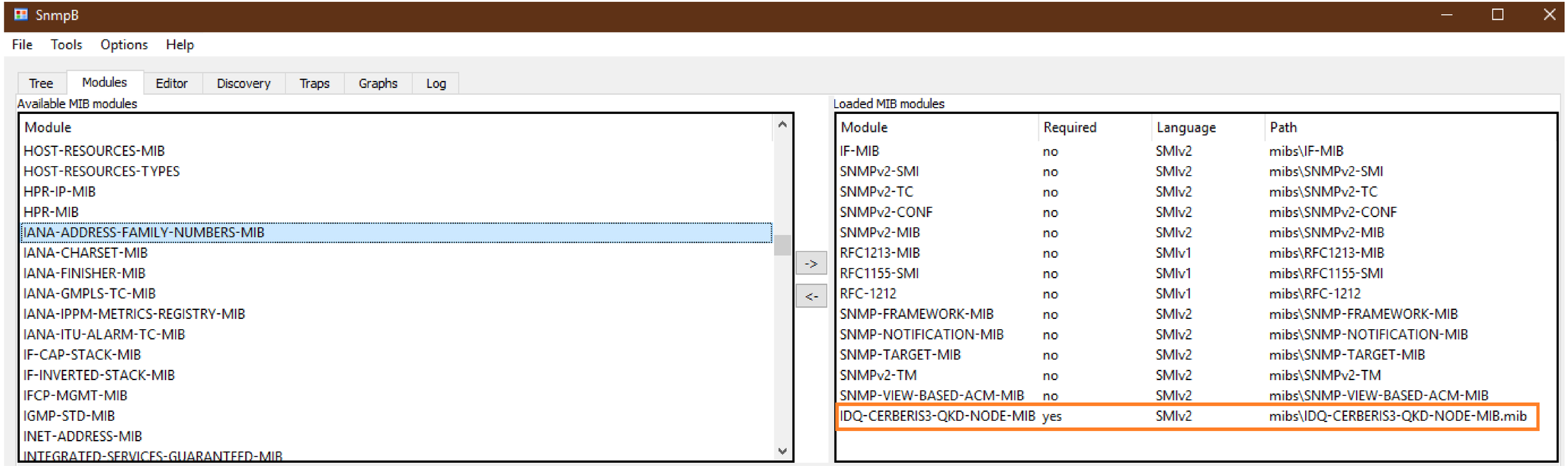
Wrong information. Do not pay attention

Move Both MIB Files!

SNMPB configuration

3. Configure MIBs on SNMPB

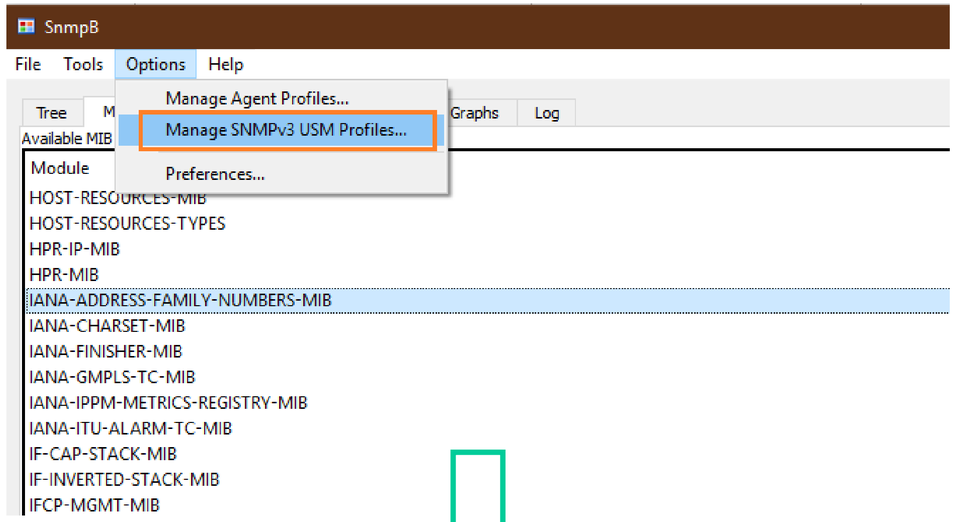
- The MIBs should appear to the right side (Loaded MIB modules)

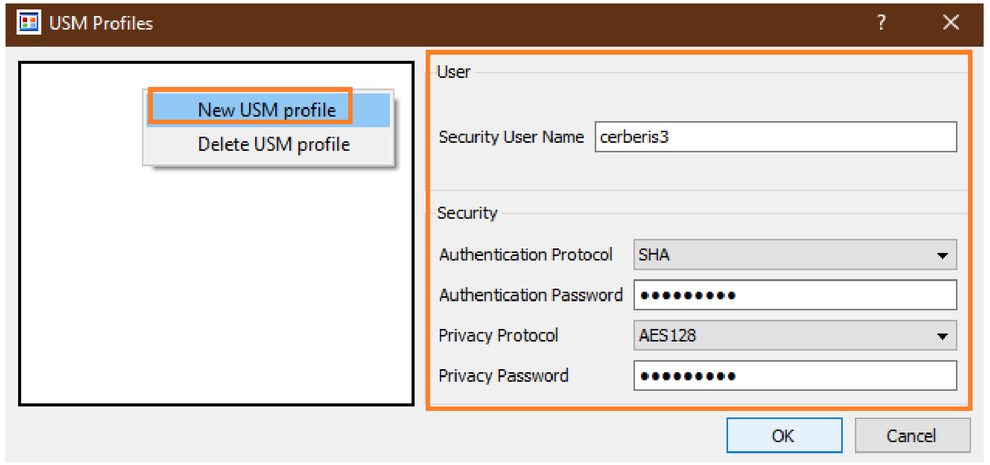


SNMPB configuration

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.

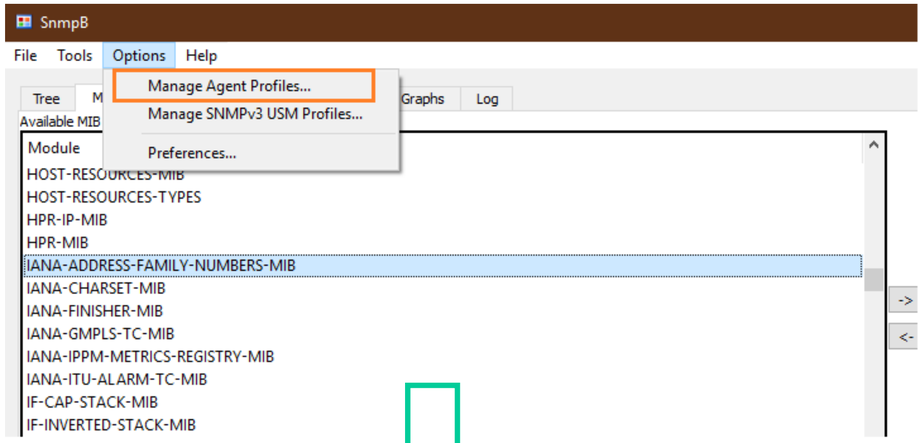


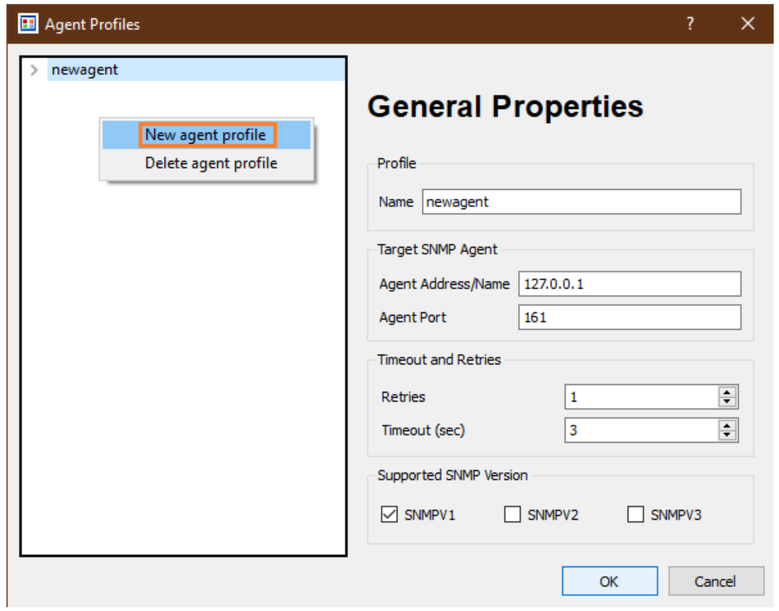


SNMPB configuration

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

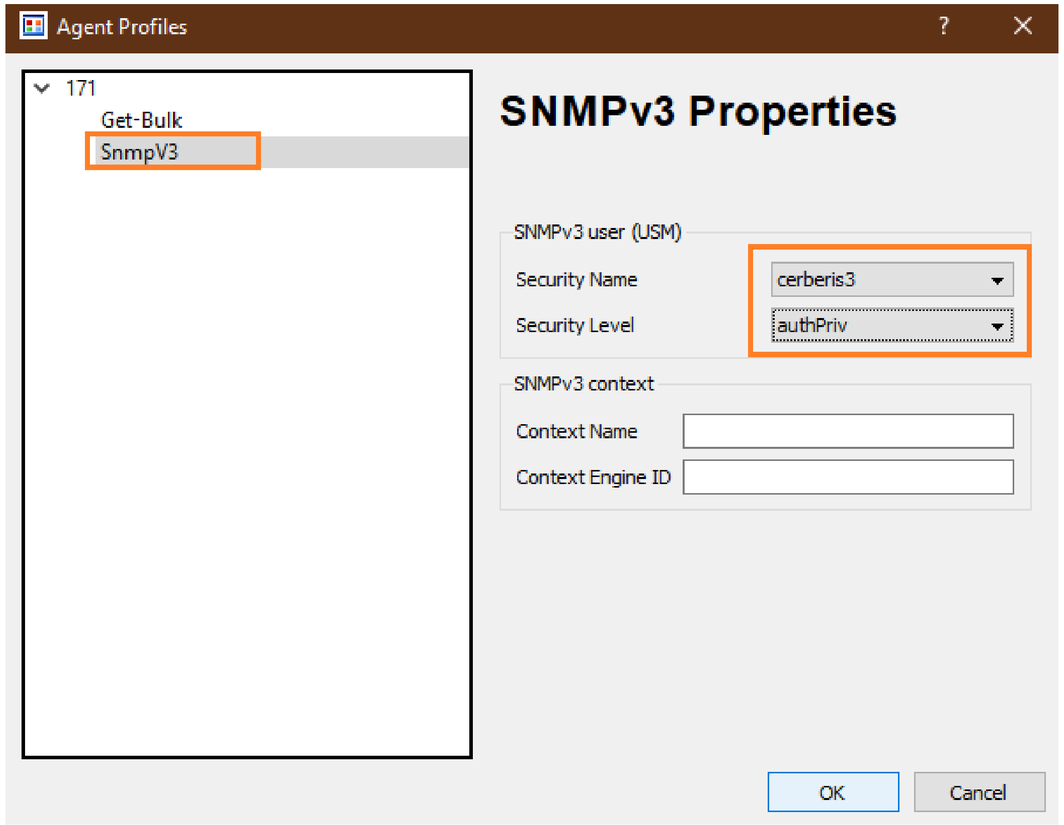




SNMPB configuration

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 QNC

```
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 QNC

Help

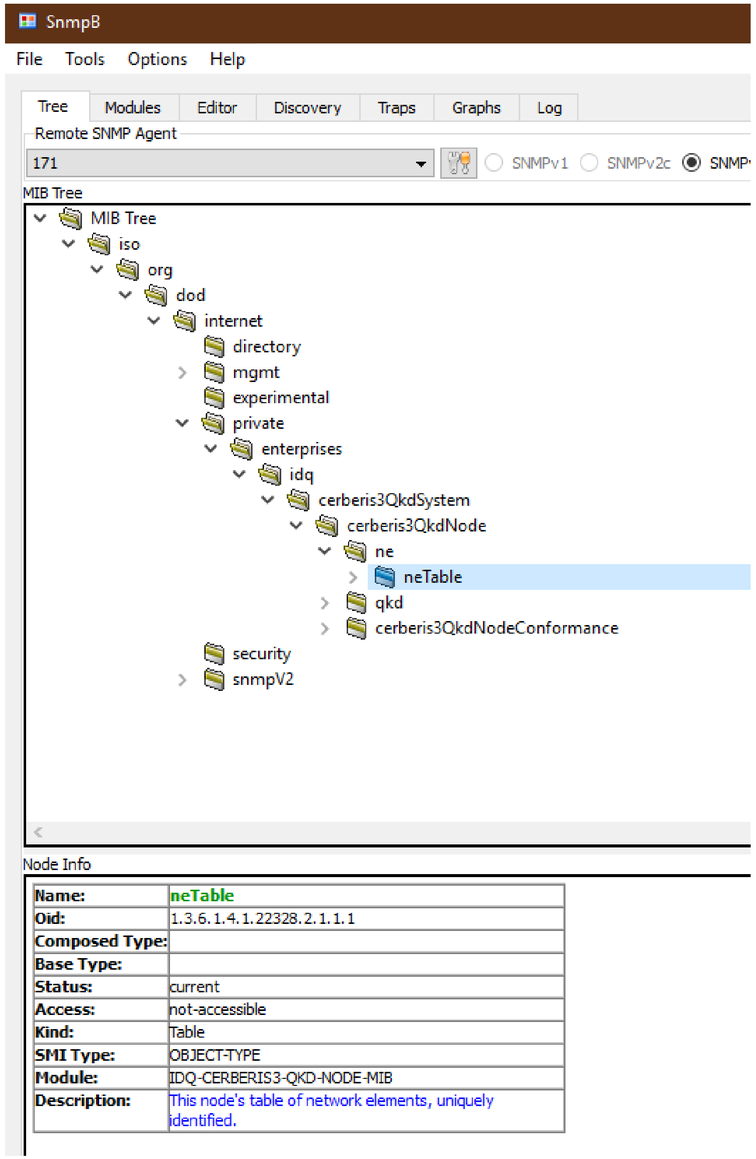
```
admin> snmp --help
manage the SNMP configuration

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>
```


ID Quantique PROPRIETARY

SNMPB configuration

7. Check if the communication is working

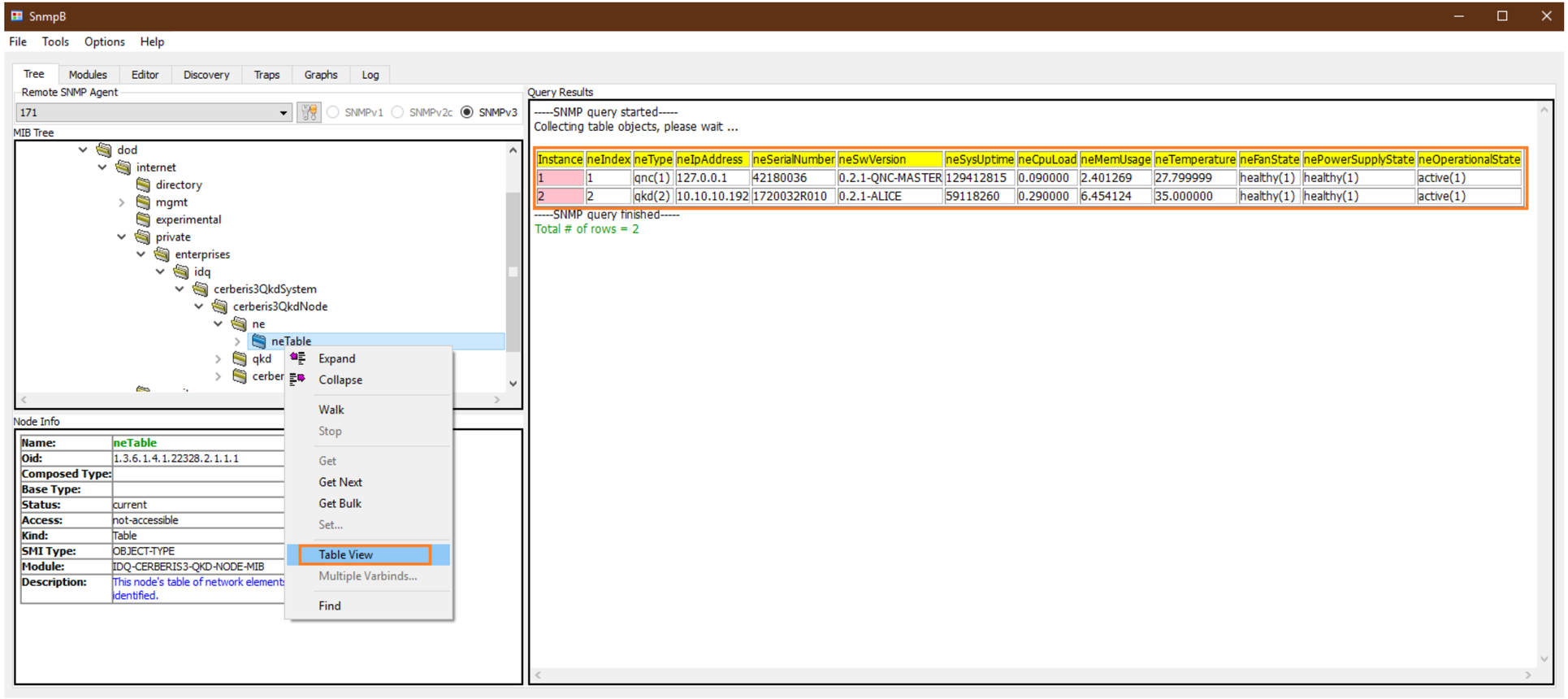
- Go to MIB Tree → iso → org → dod → internet → private → enterprise → idq → cerberis3QkdSystem



SNMPB configuration

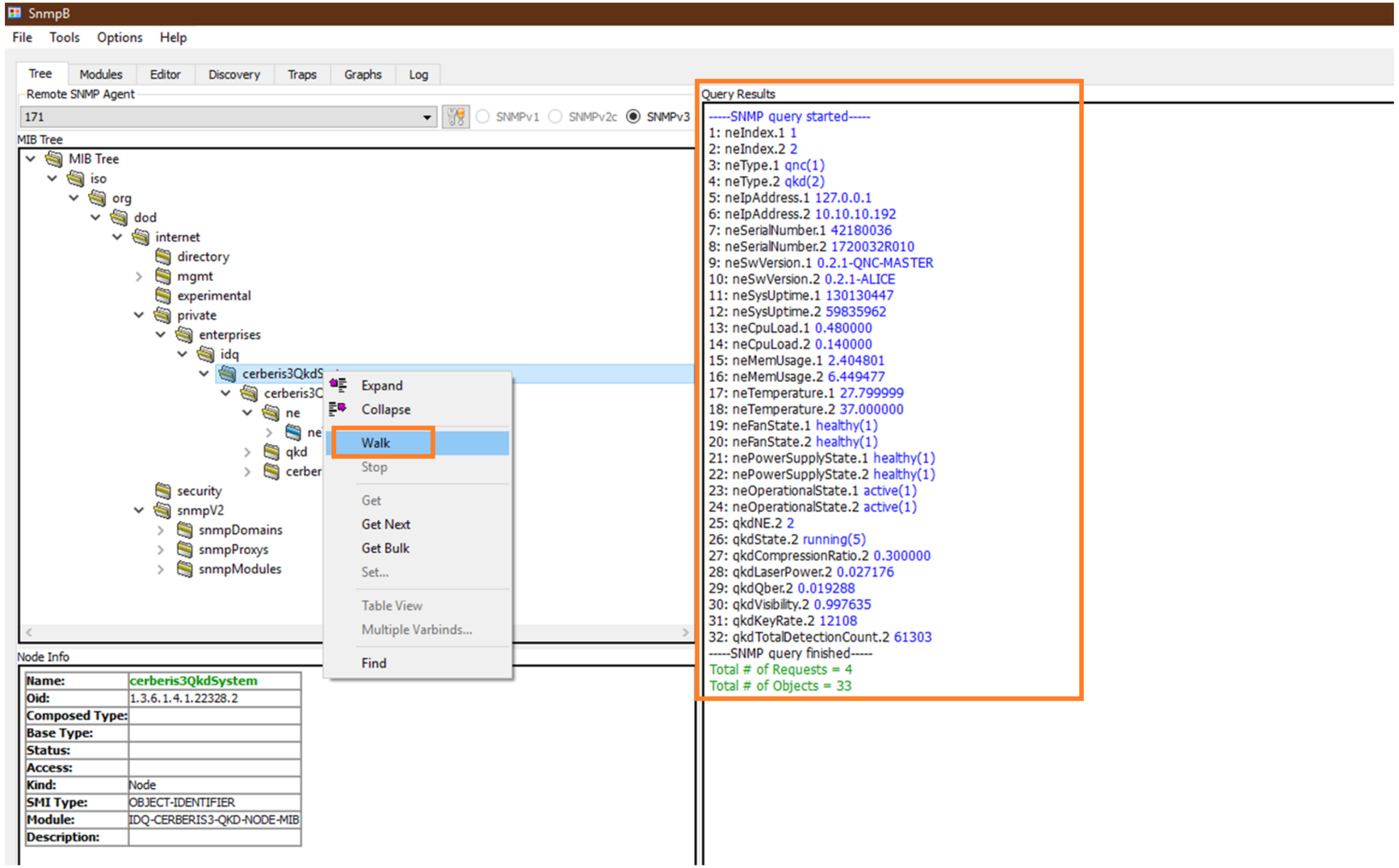
7. Check if the communication is working

- Develop 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

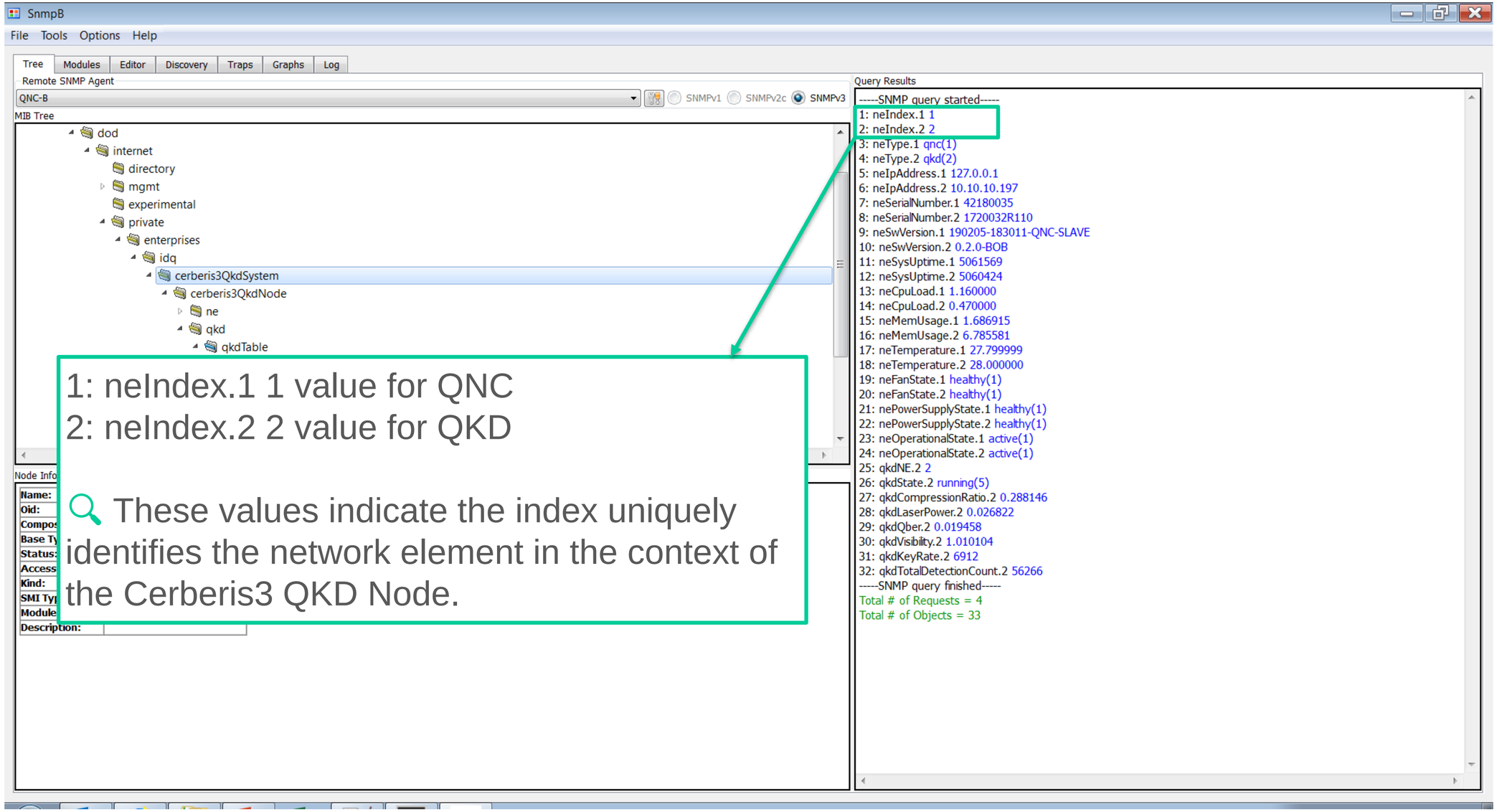
- Develop “cerberis3QkdSystem”, do a right click and then press “Walk”



SNMP Monitoring – Get values

- 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.

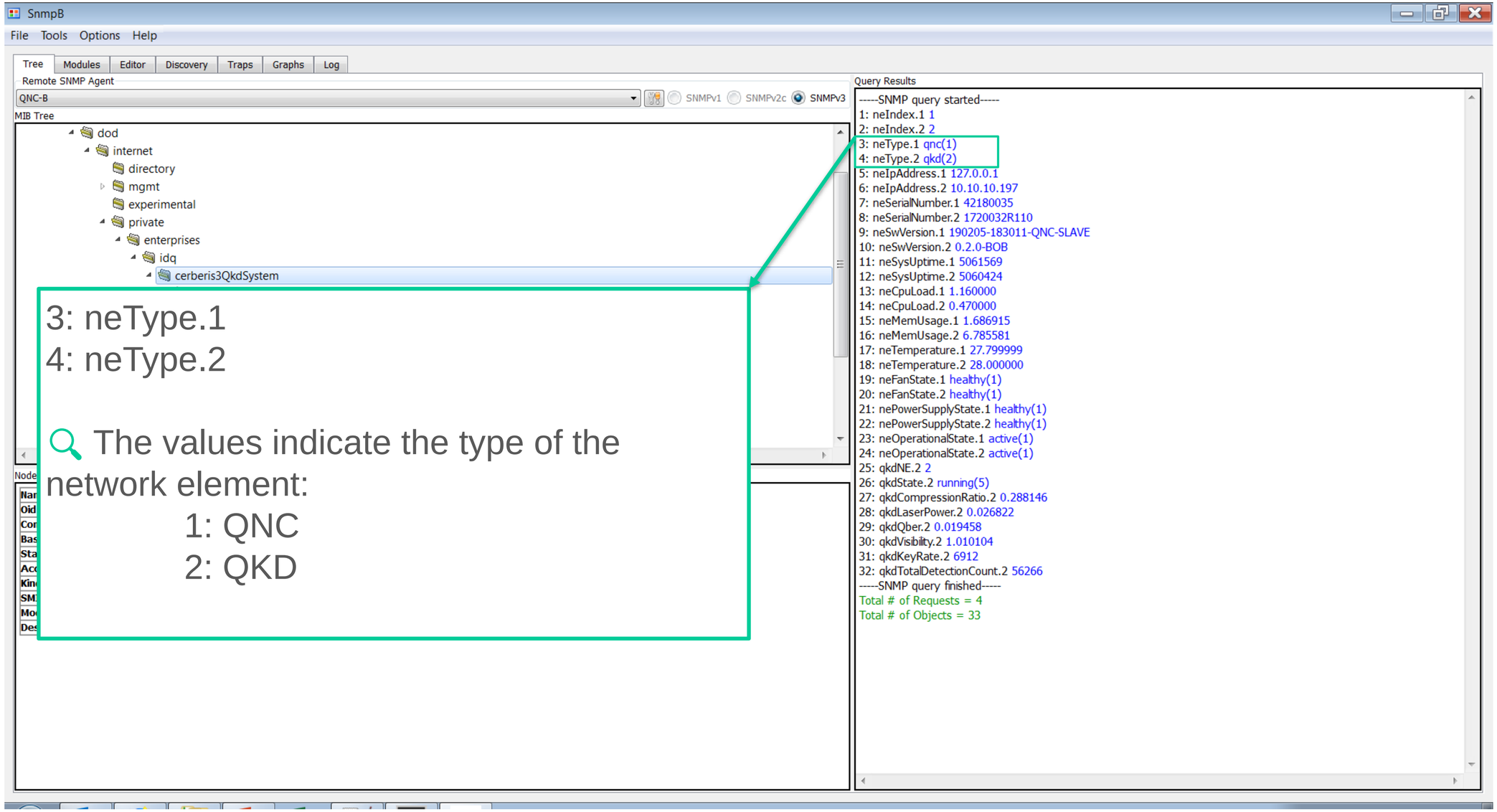
SNMP Monitoring – Values Explanations



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

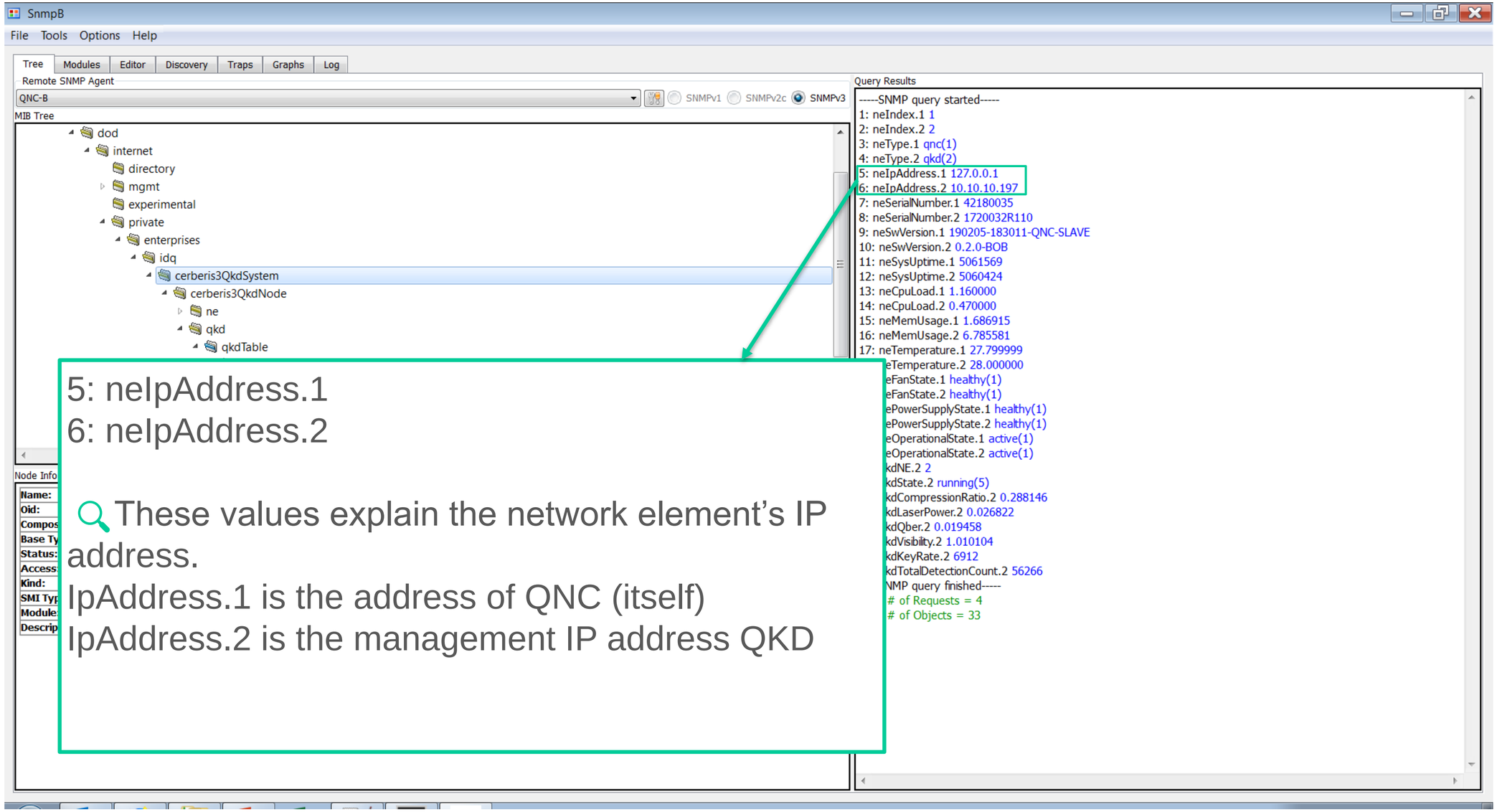


3: neType.1
4: neType.2

🔍 The values indicate the type of the network element:

1: QNC
2: QKD

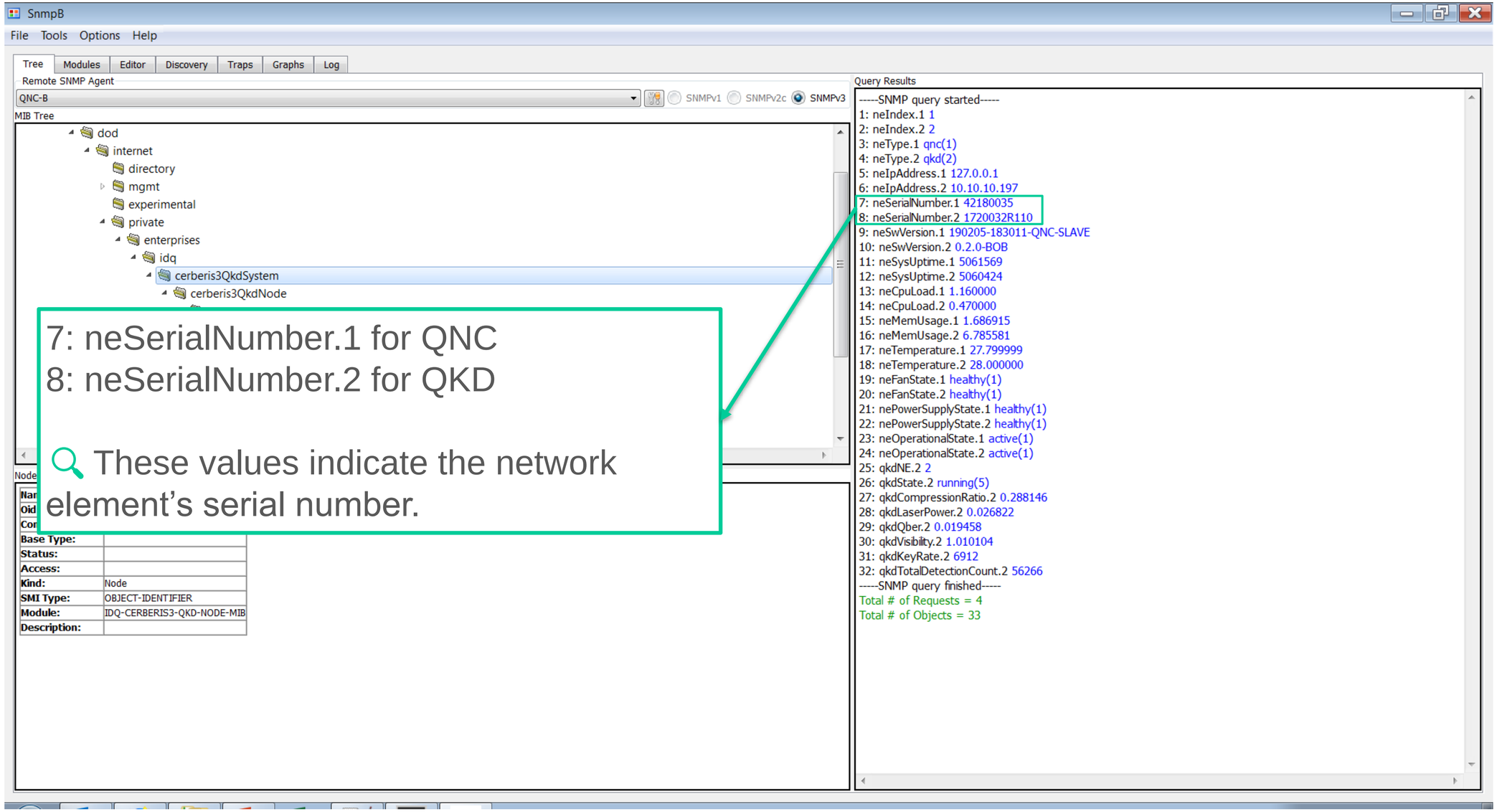
SNMP Monitoring – Values Explanations



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

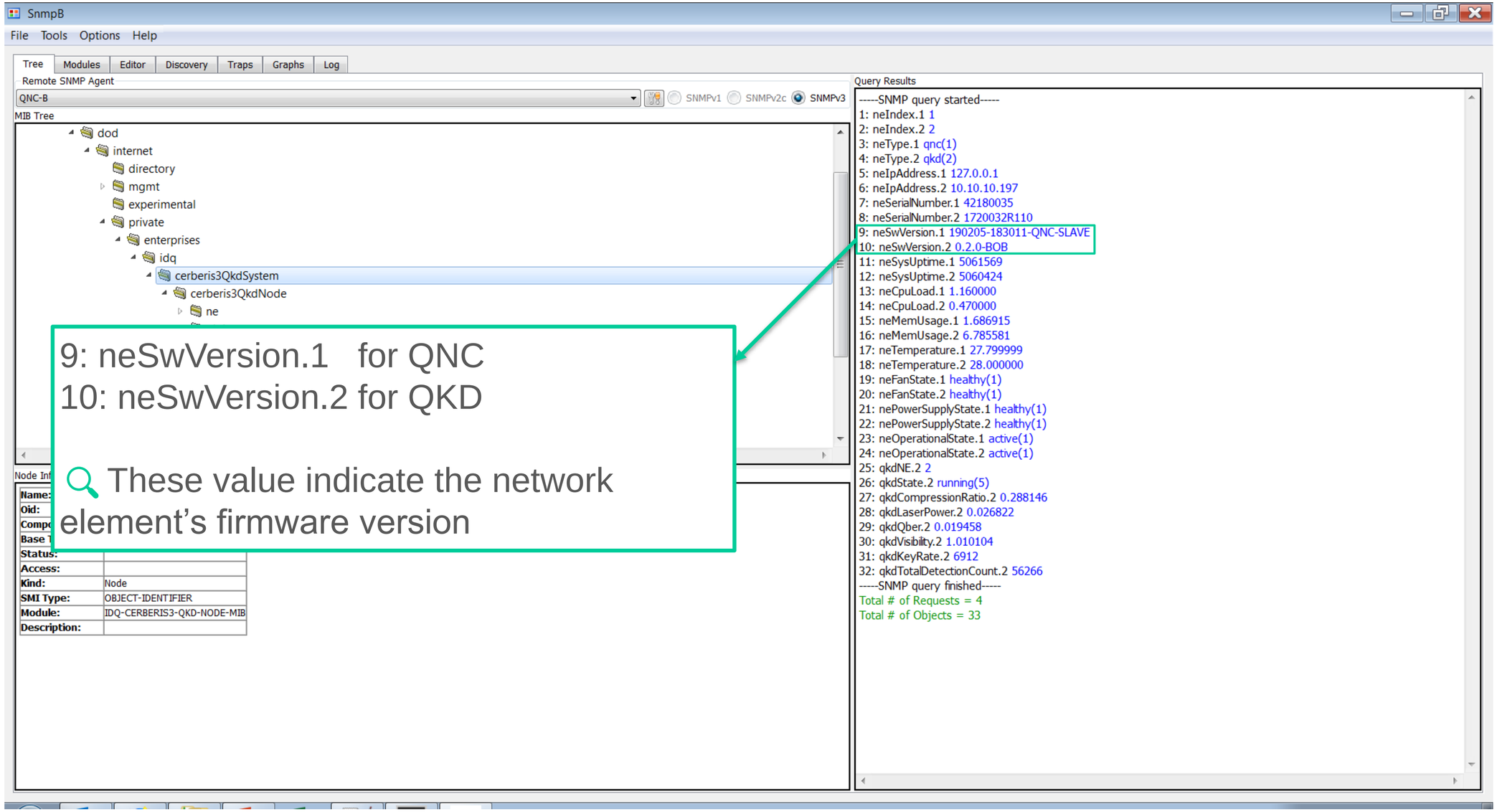
SNMP Monitoring – Values Explanations



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

🔍 These values indicate the network element's serial number.

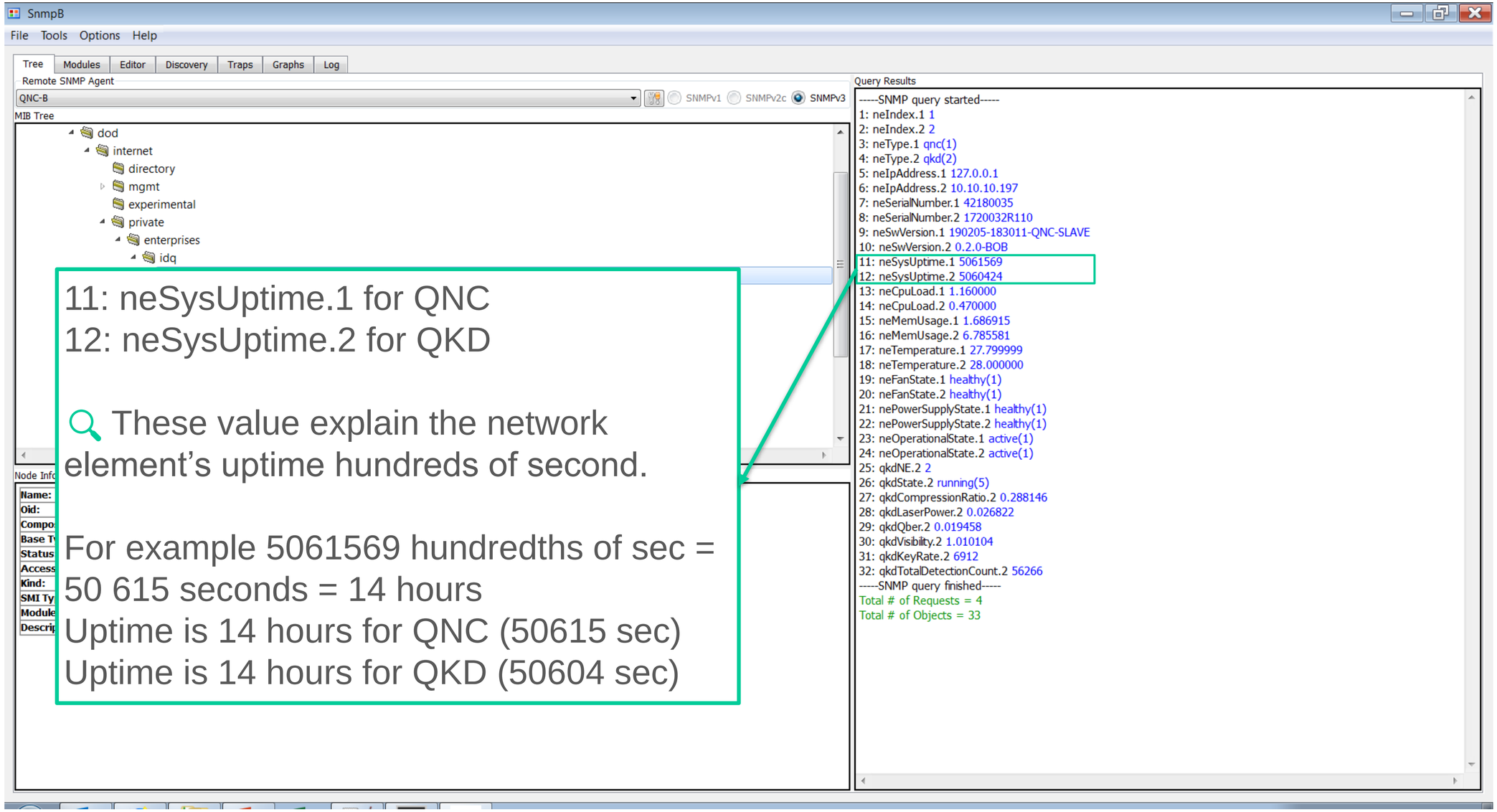
SNMP Monitoring – Values Explanations



9: neSwVersion.1 for QNC
10: neSwVersion.2 for QKD

These value indicate the network element's firmware version

SNMP Monitoring – Values Explanations

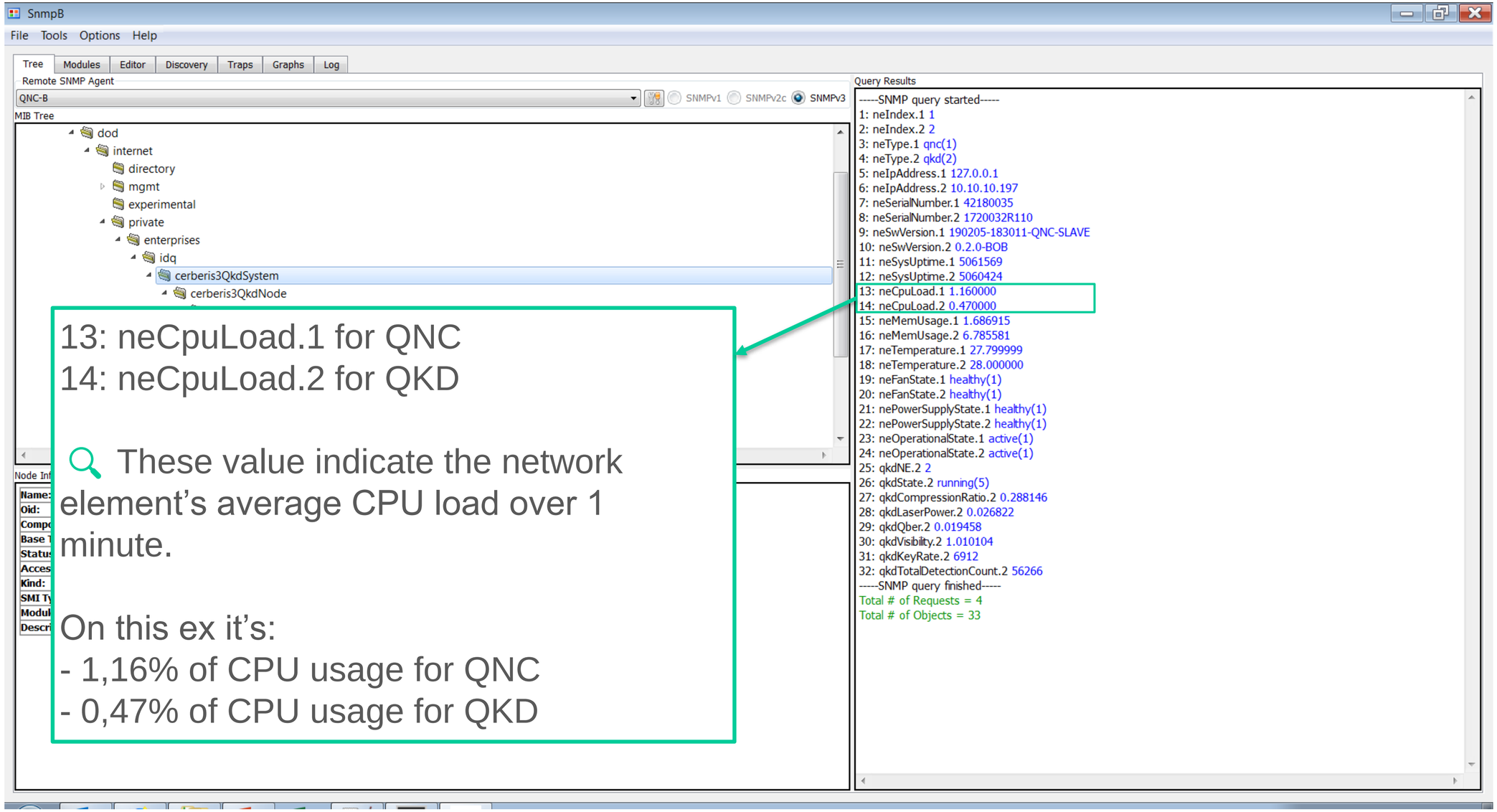


11: neSysUptime.1 for QNC
12: neSysUptime.2 for QKD

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

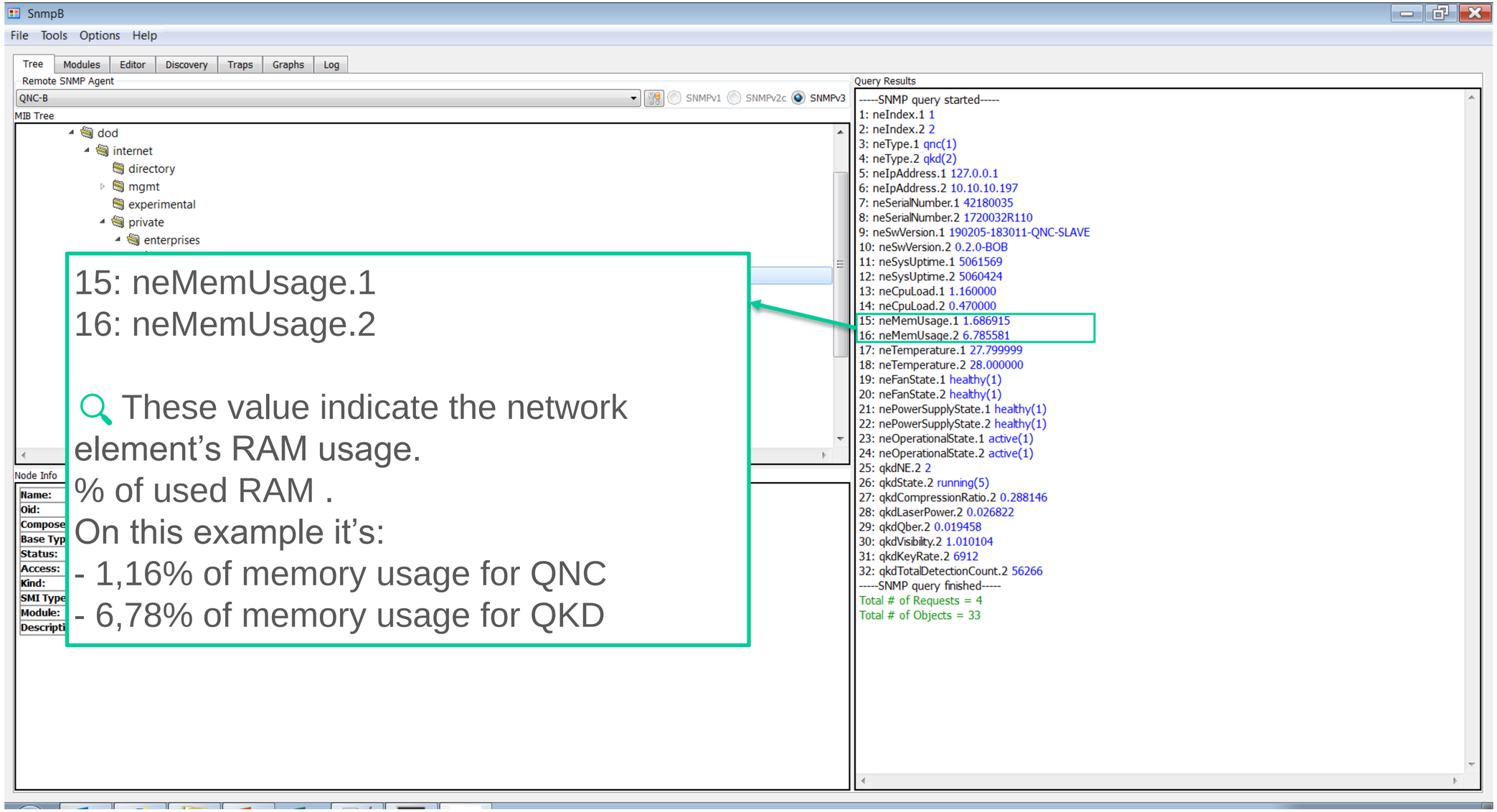


13: neCpuLoad.1 for QNC
14: neCpuLoad.2 for QKD

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

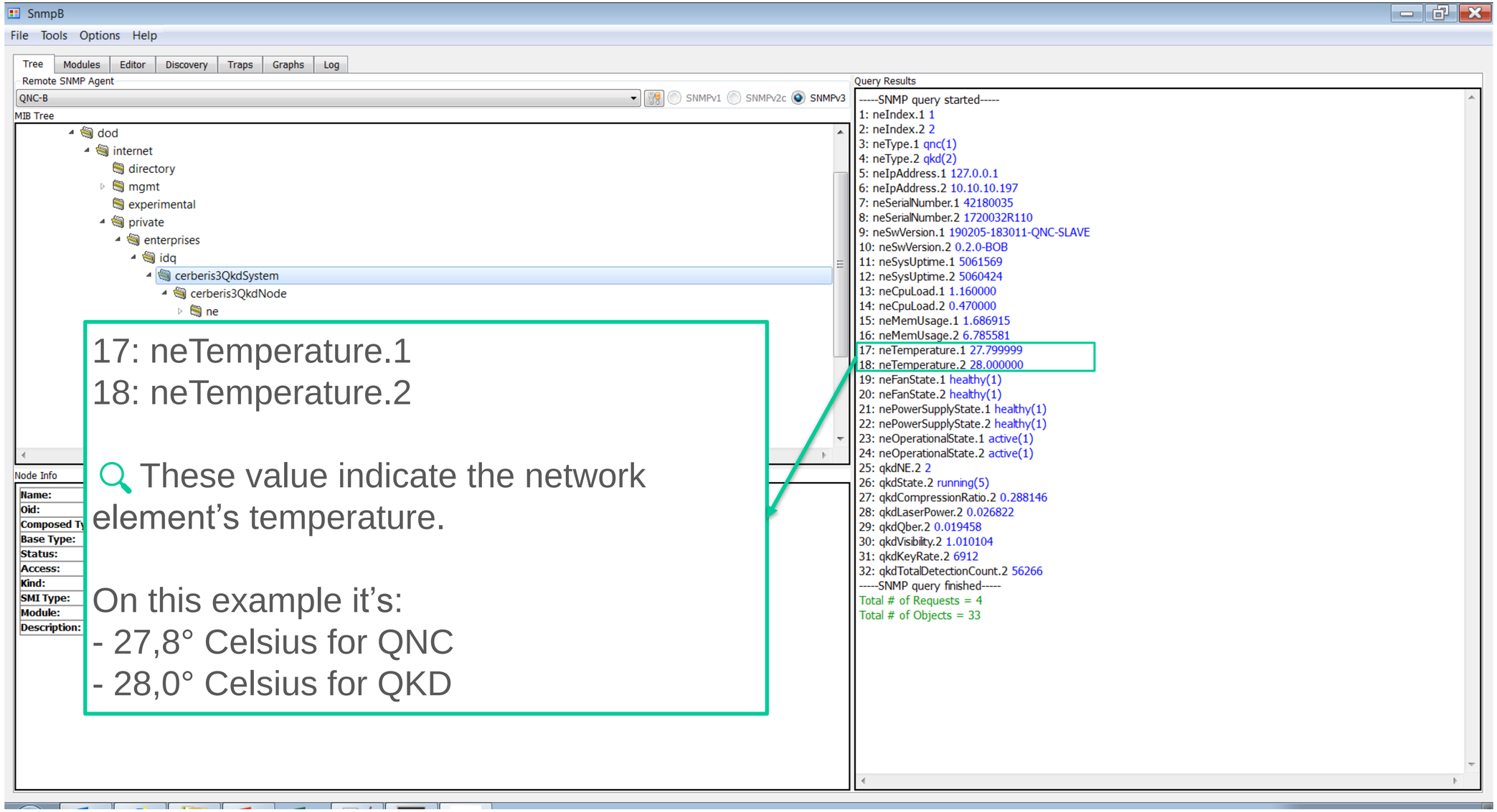
SNMP Monitoring – Values Explanations



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

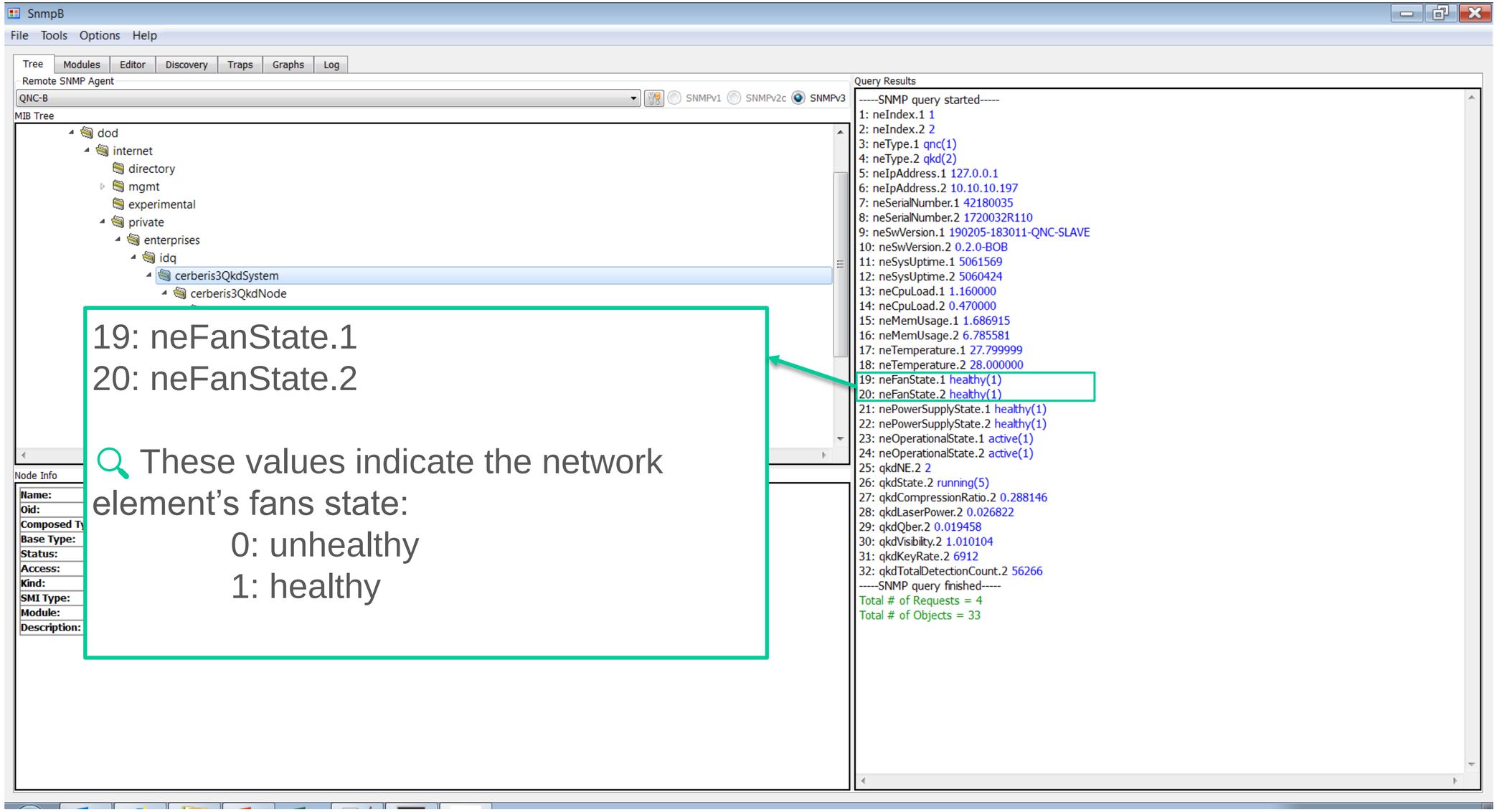


17: neTemperature.1
18: neTemperature.2

These value indicate the network element's temperature.

On this example it's:
- 27,8° Celsius for QNC
- 28,0° Celsius for QKD

SNMP Monitoring – Values Explanations

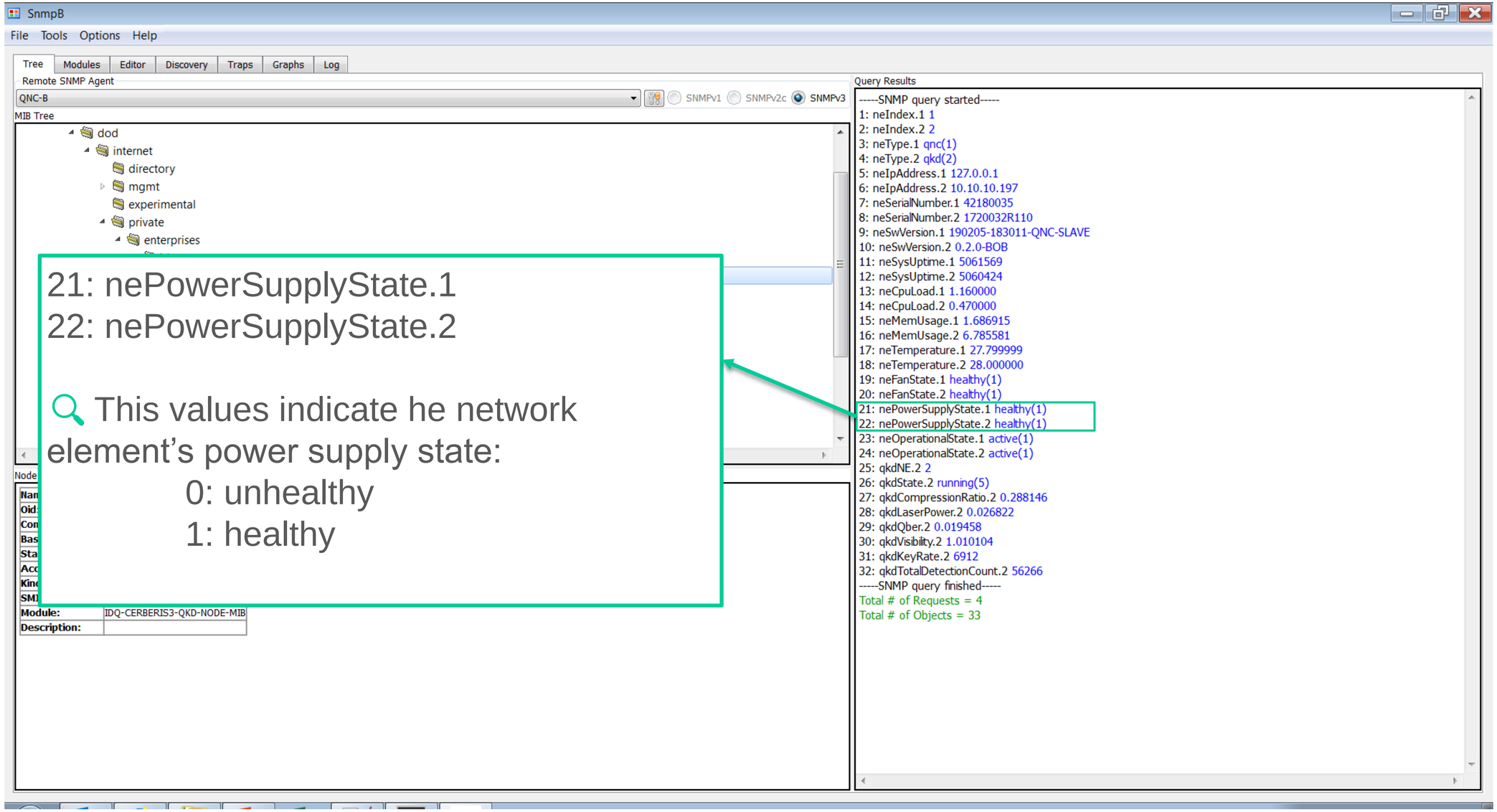


19: neFanState.1
20: neFanState.2

These values indicate the network element's fans state:

0: unhealthy
1: healthy

SNMP Monitoring – Values Explanations

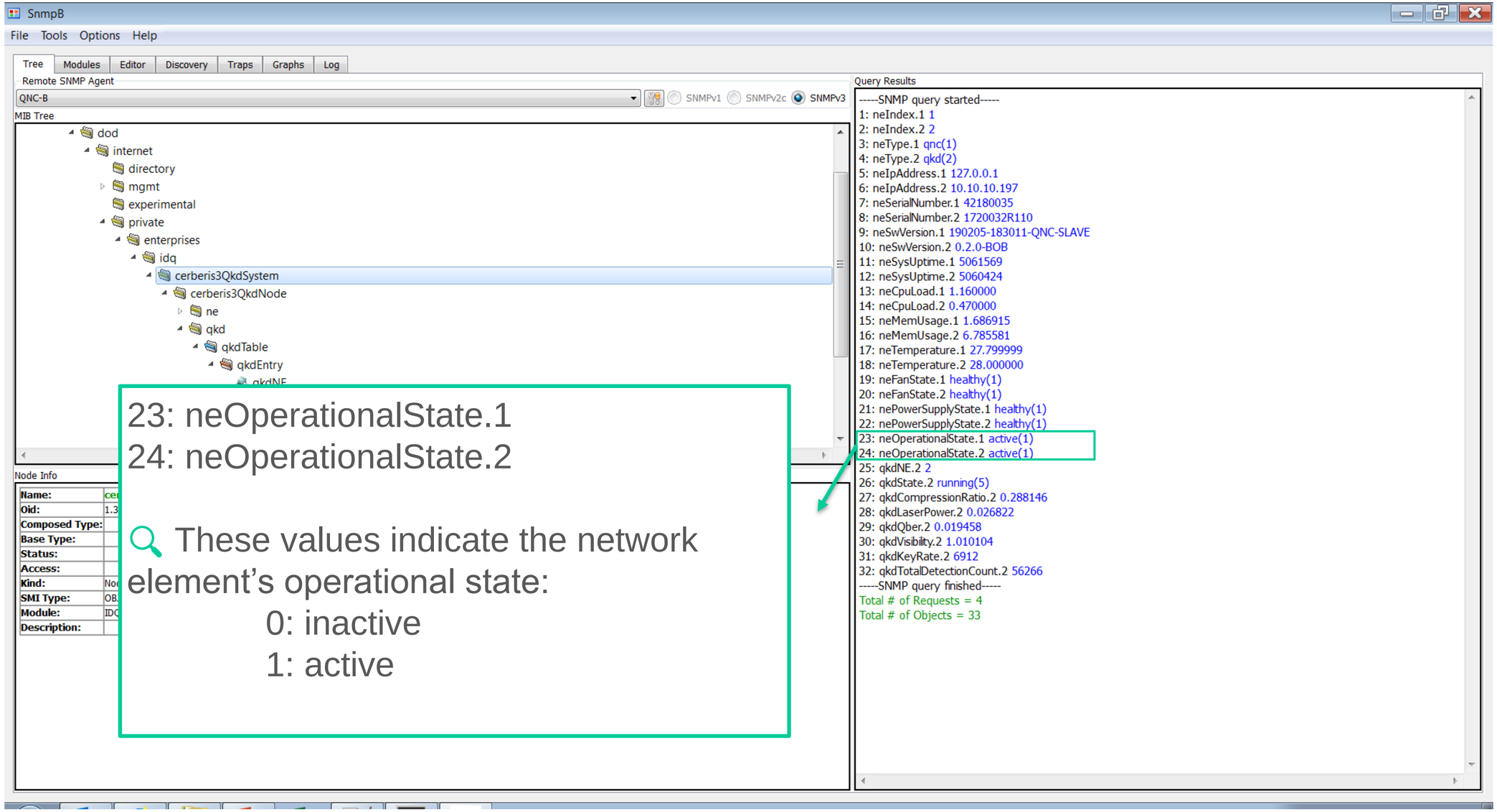


21: nePowerSupplyState.1
22: nePowerSupplyState.2

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

0: unhealthy
1: healthy

SNMP Monitoring – Values Explanations

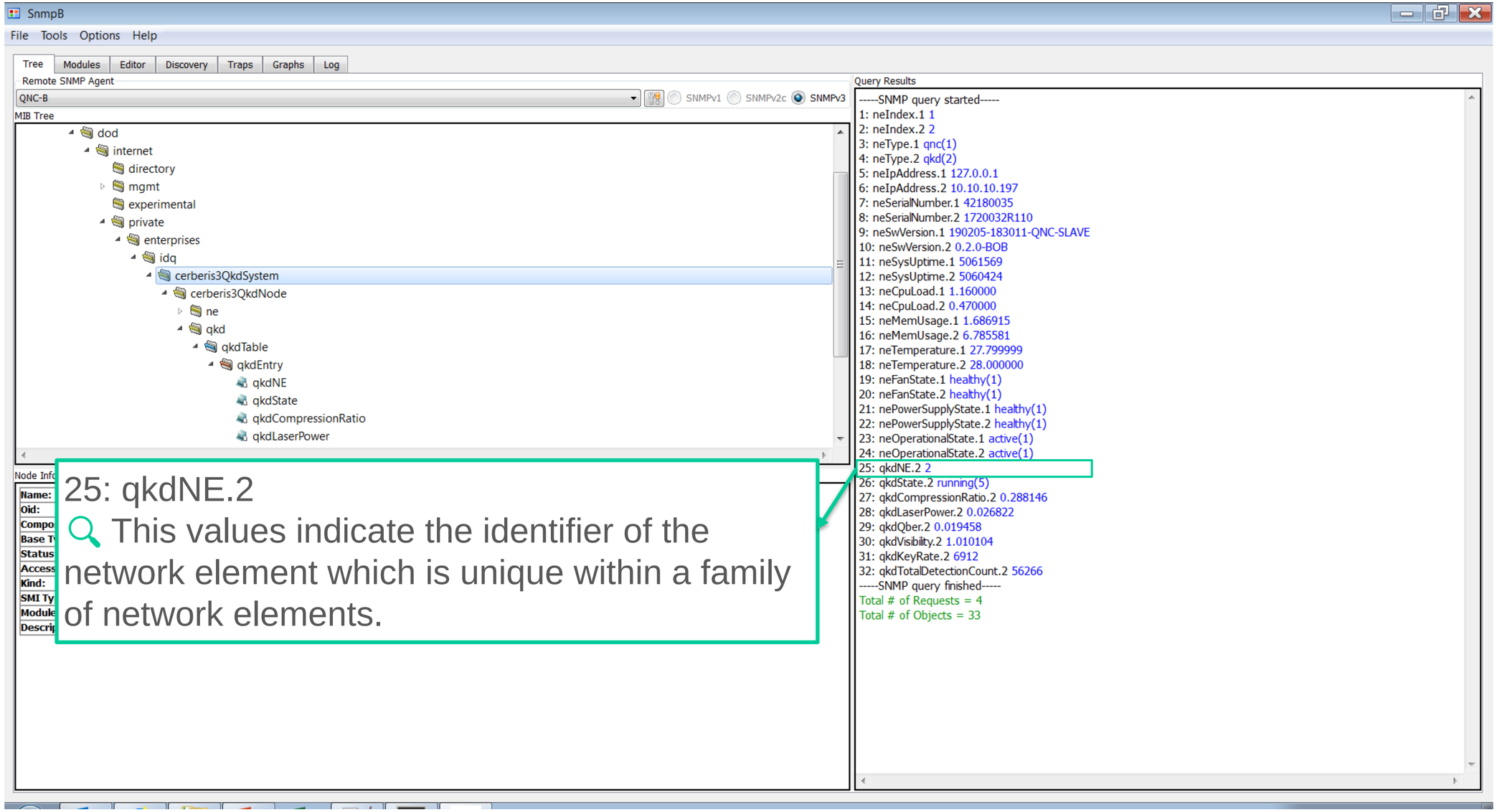


23: neOperationalState.1
24: neOperationalState.2

These values indicate the network element's operational state:

0: inactive
1: active

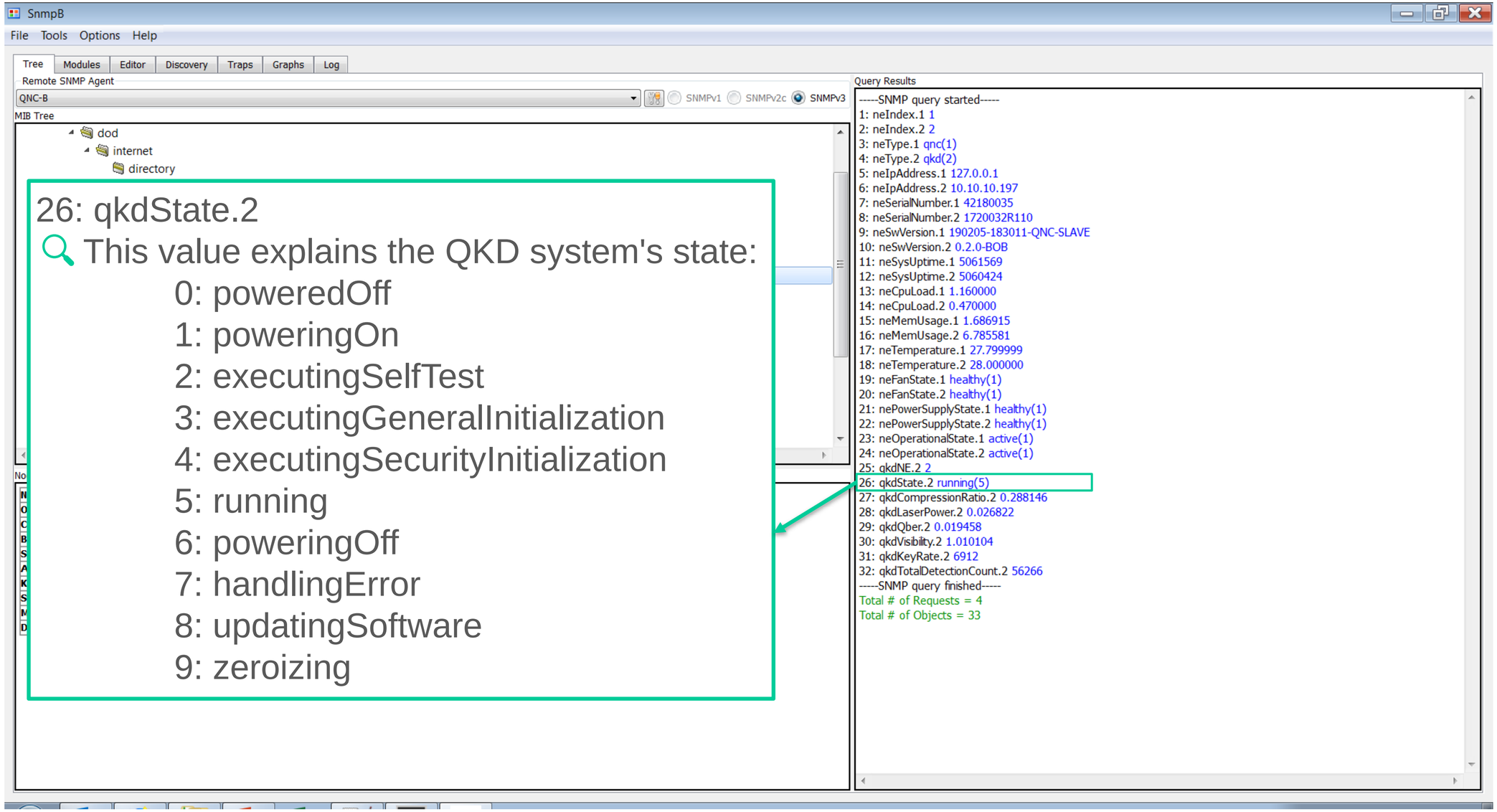
SNMP Monitoring – Values Explanations



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



26: qkdState.2

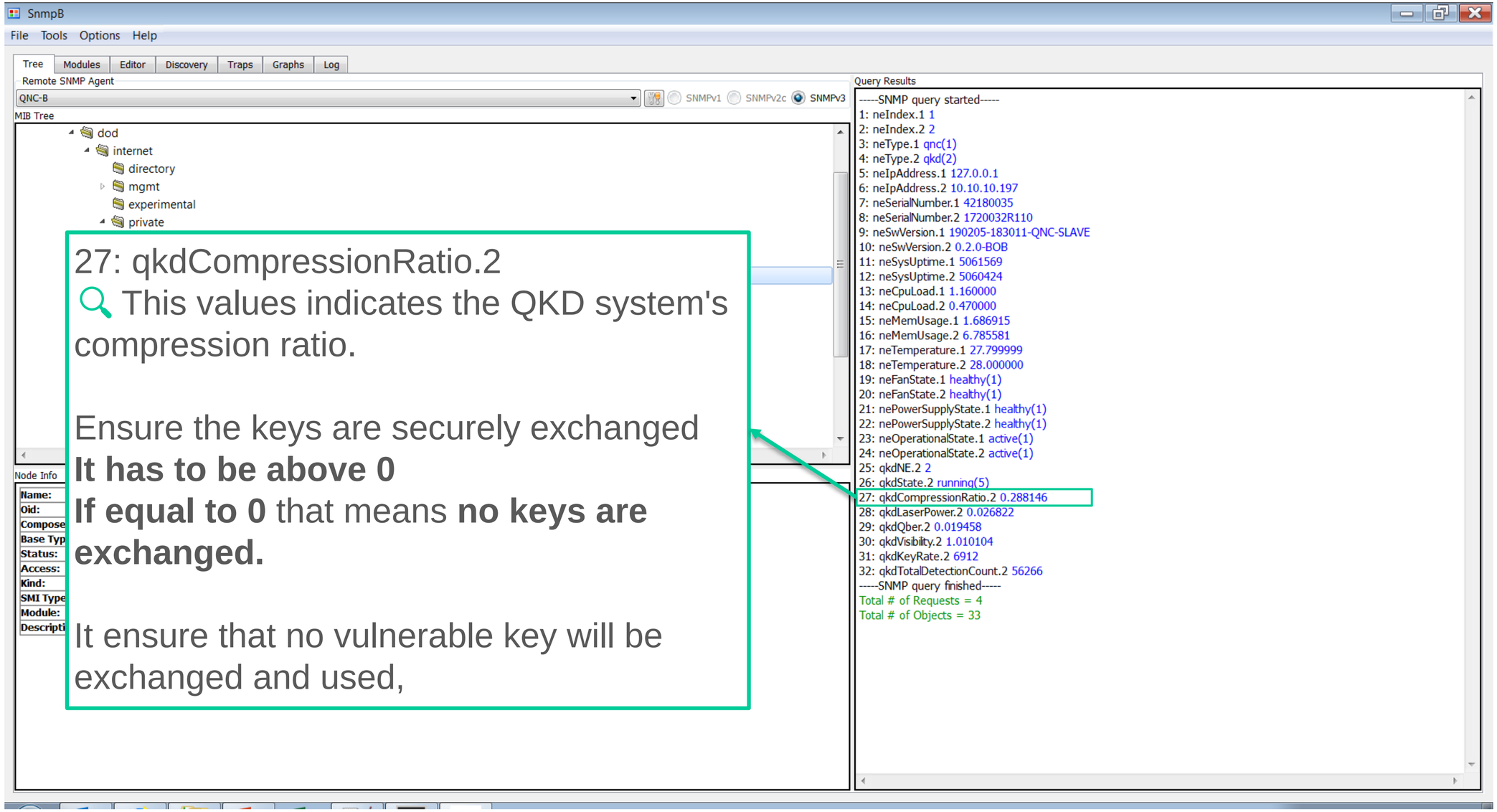
This value explains the QKD system's state:

- 0: poweredOff
- 1: poweringOn
- 2: executingSelfTest
- 3: executingGeneralInitialization
- 4: executingSecurityInitialization
- 5: running
- 6: poweringOff
- 7: handlingError
- 8: updatingSoftware
- 9: zeroizing

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.

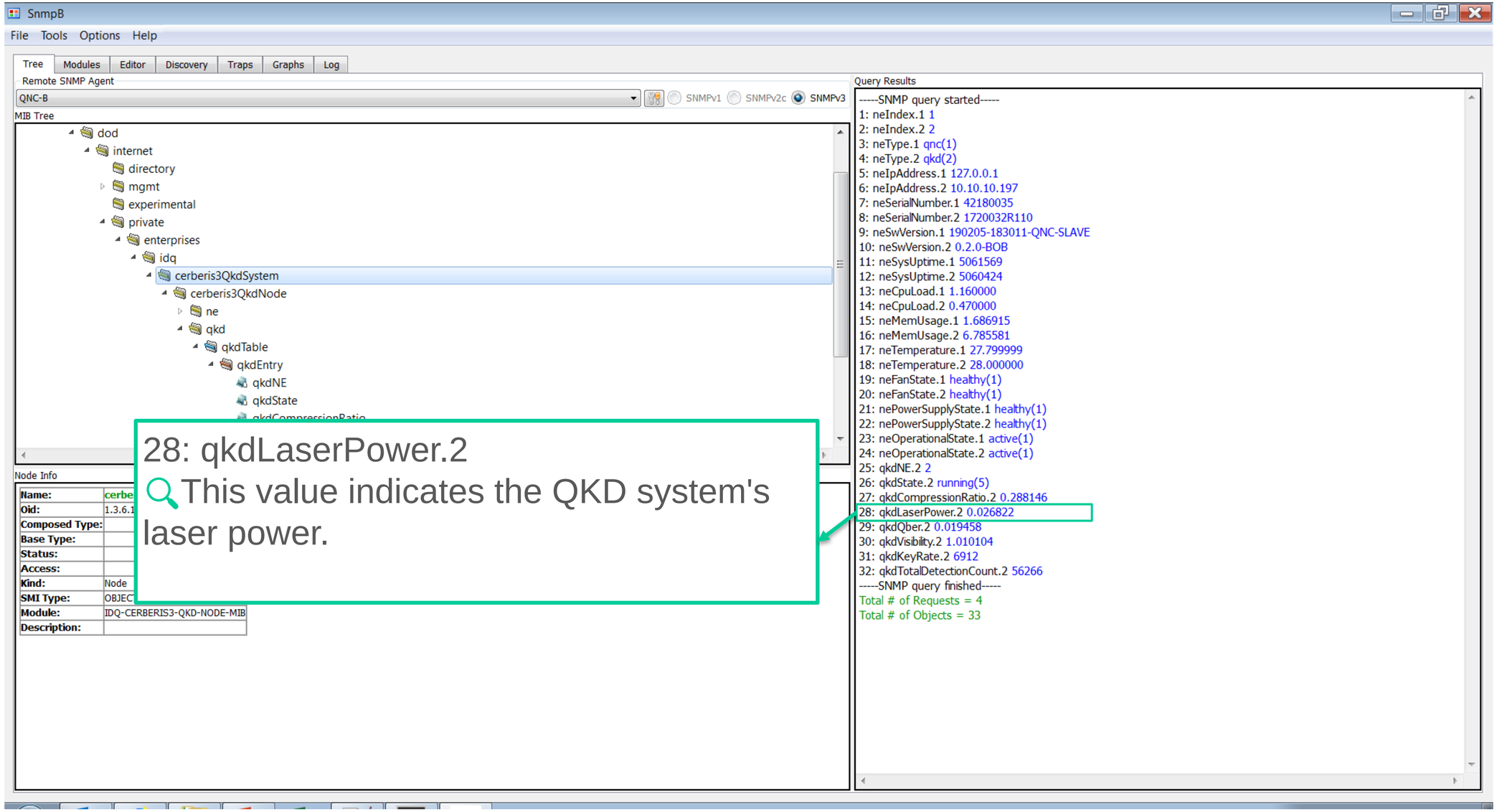
Ensure the keys are securely exchanged

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,

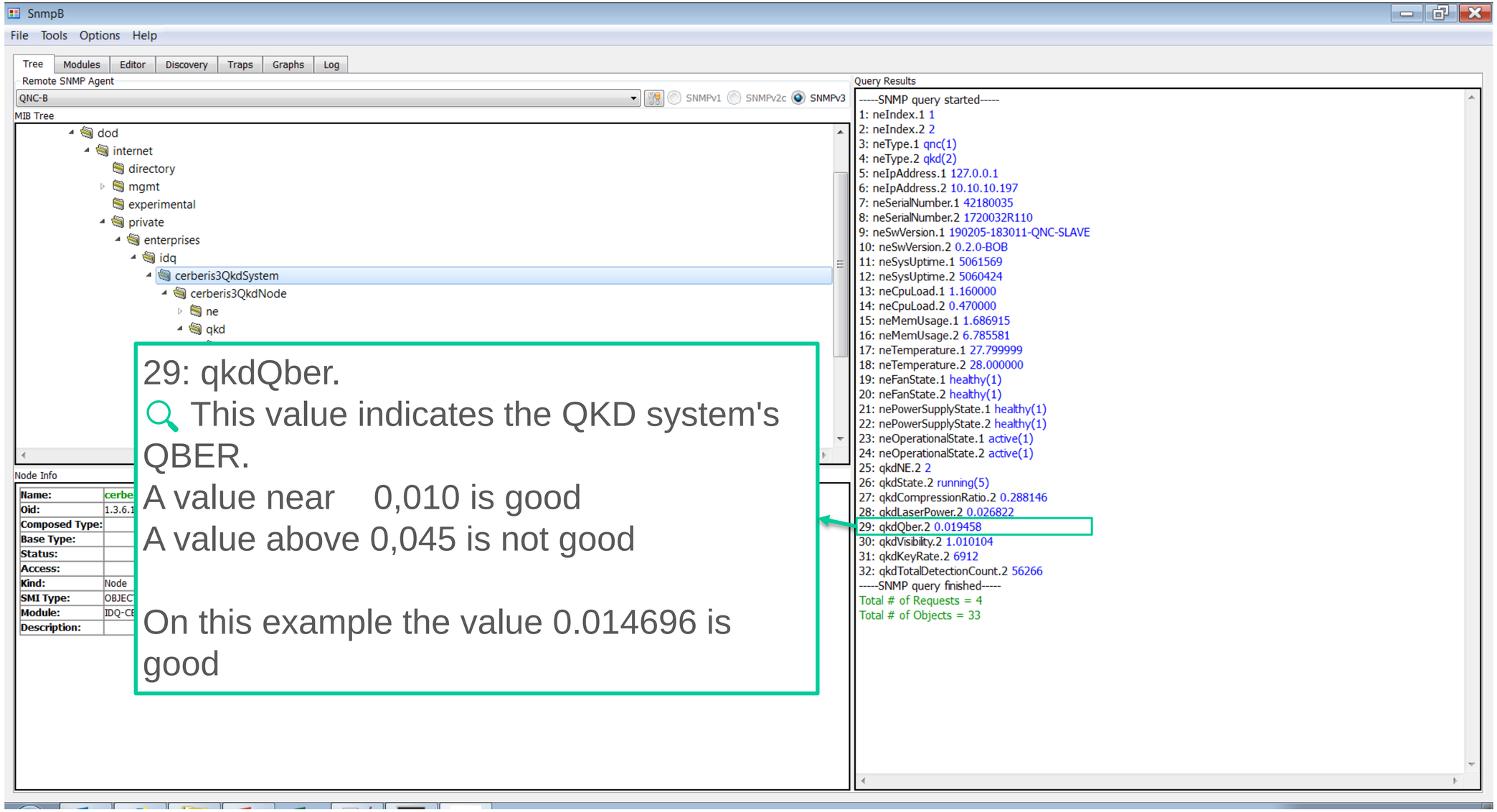
SNMP Monitoring – Values Explanations



28: qkdLaserPower.2

This value indicates the QKD system's laser power.

SNMP Monitoring – Values Explanations



29: qkdQber.

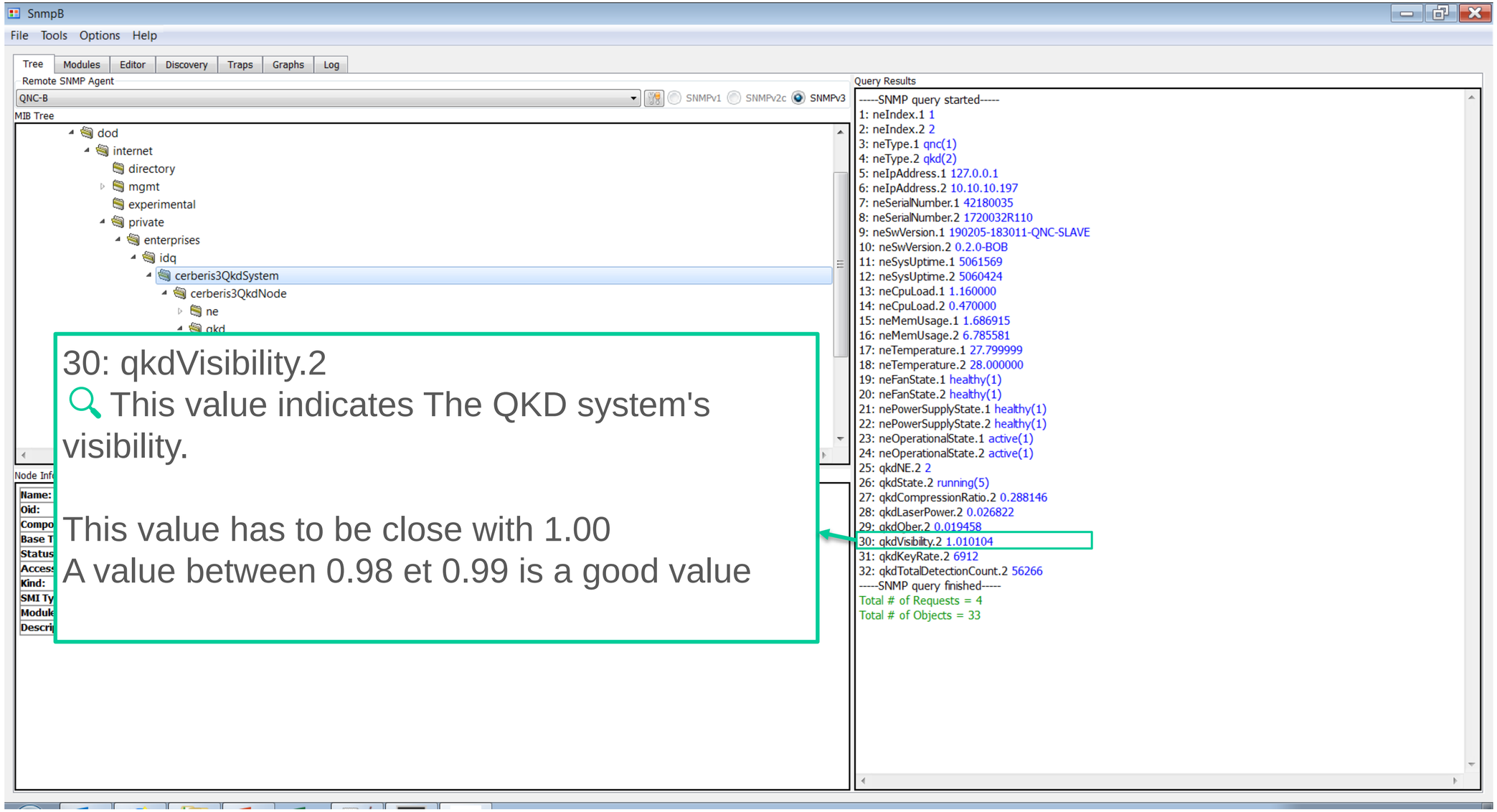
This value indicates the QKD system's QBER.

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



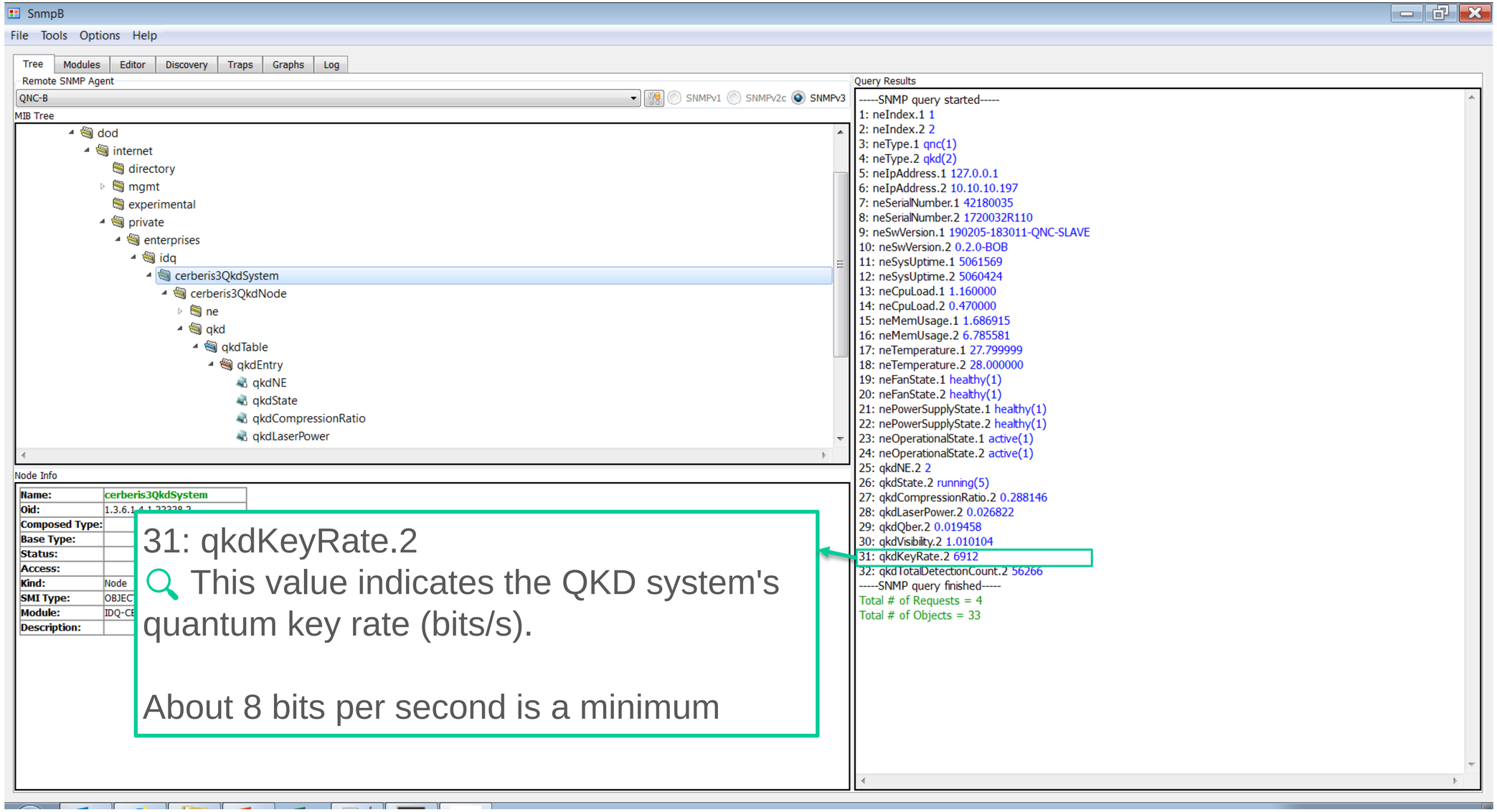
30: qkdVisibility.2

🔍 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

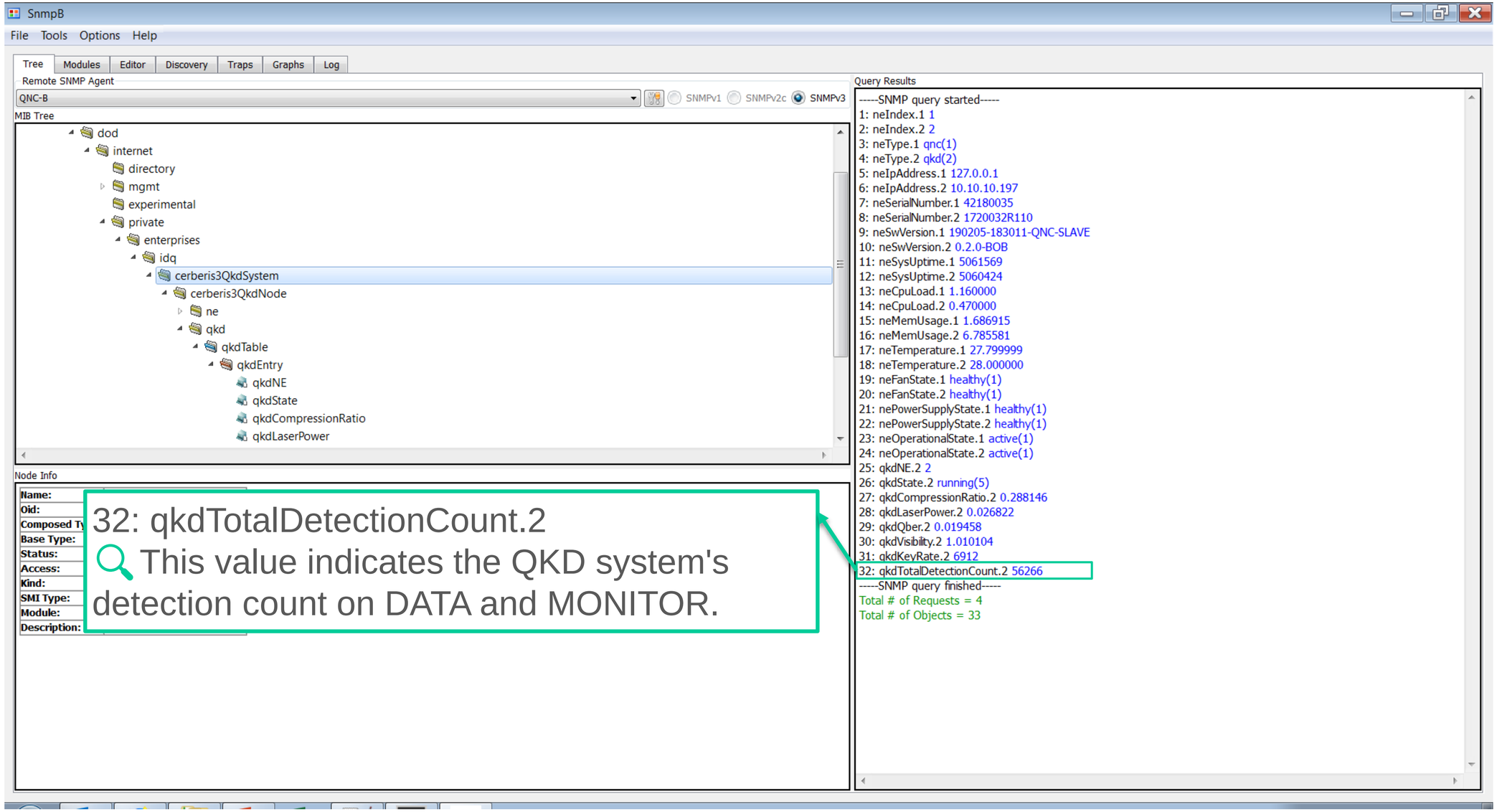


31: qkdKeyRate.2

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

About 8 bits per second is a minimum

SNMP Monitoring – Values Explanations



SNMP Monitoring – Summarize

SWISS QUANTUM

Summarize

- 2 values **are important to check:**
  - **Visibility** (it has to be close with 1)
  - **QBER** (lower than 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

IDQ

ETSI client

ETSI

World Class Standards

IDQ

FROM VISION TO TECHNOLOGY

ETSI client

- **ETSI** is the **protocol to connect third party** devices to the QNCs
- Usually the third party devices are:
    - **Encryptors**
    - **HSM**
    - Encryptors **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 QNC frontpane port

- Connect your ETSI client to the management network

ETSI client

You have 2 choices:

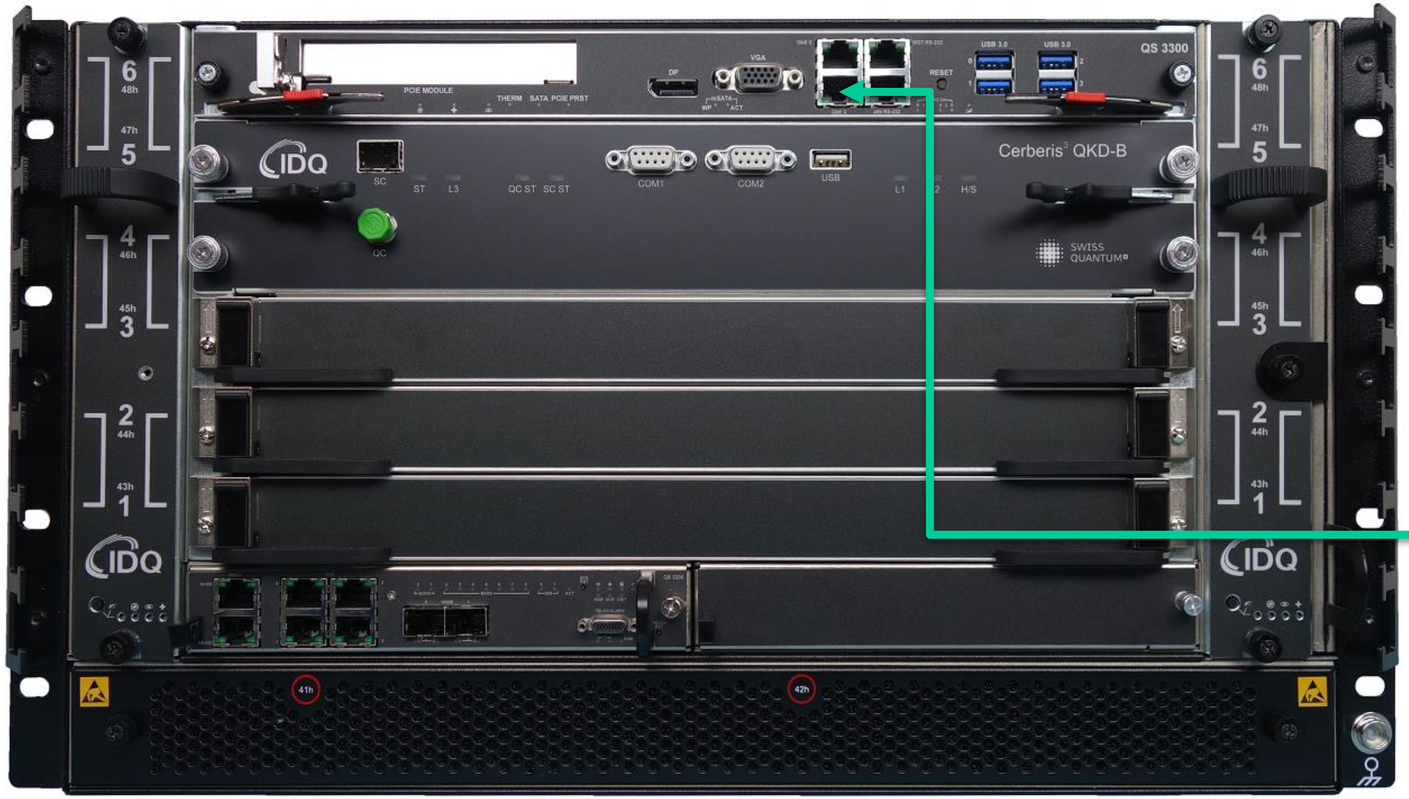
- **Connect your ETSI client to the QNC frontpane port**

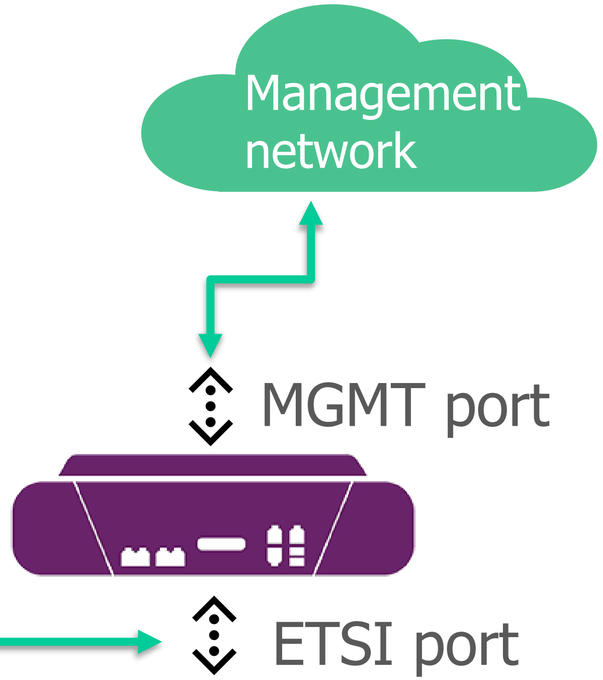
- Connect your ETSI client to the management network

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 simulator

You have 2 choices:

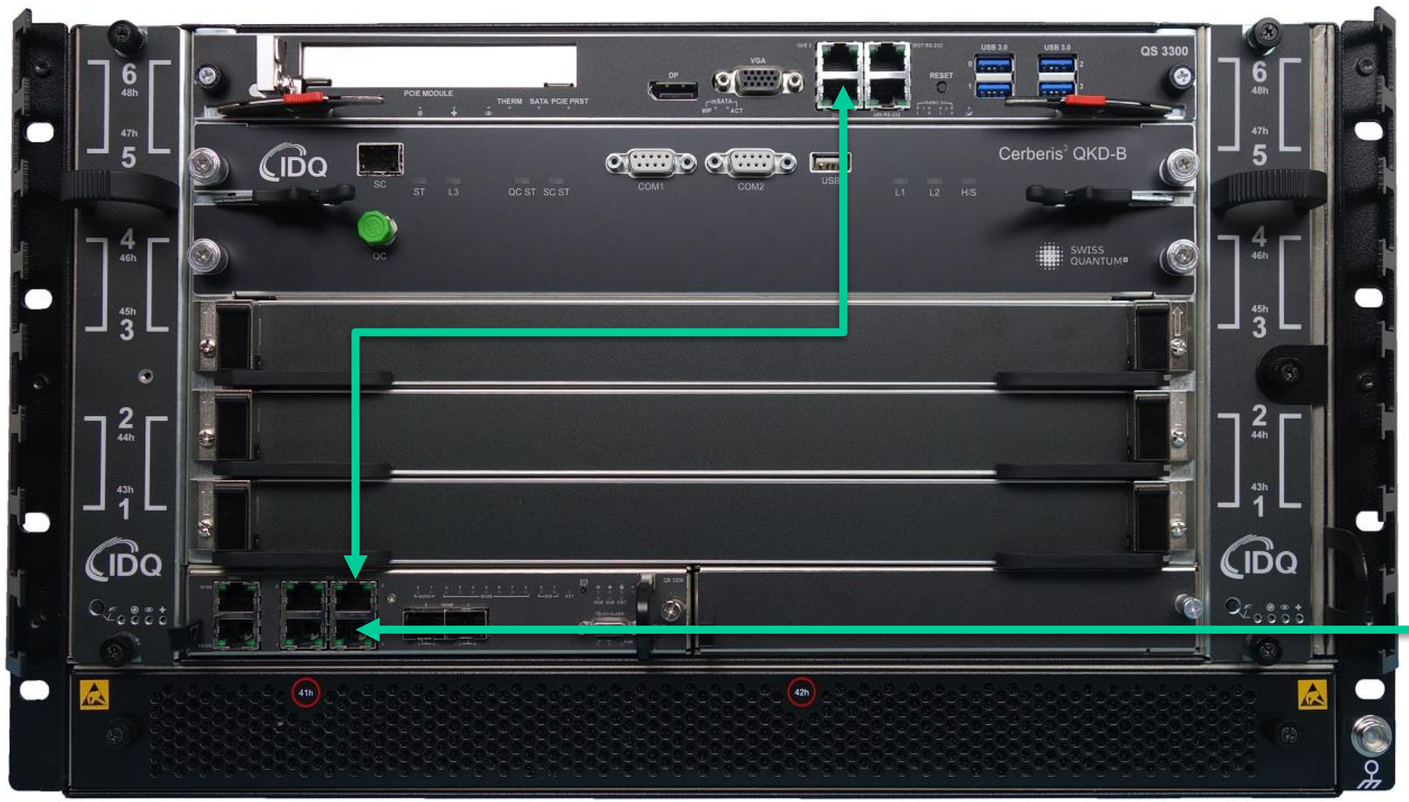
- Connect your ETSI client to the QNC frontpane port

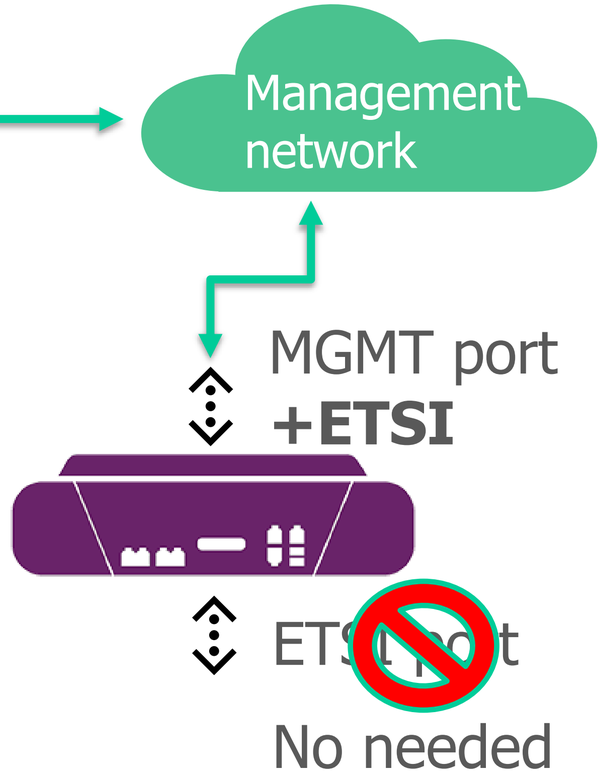
- **Connect your ETSI client to the management network**

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

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

SWISS QUANTUM

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 QNCs:
    - ❑ .key file (private key)
    - ❑ .cert file (public key)

IDQ

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

ETSI client

Understand the ETSI TLS process

Client

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

Encryptor

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

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

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

ETSI client

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

QNC

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

ClientCA.cert.pem

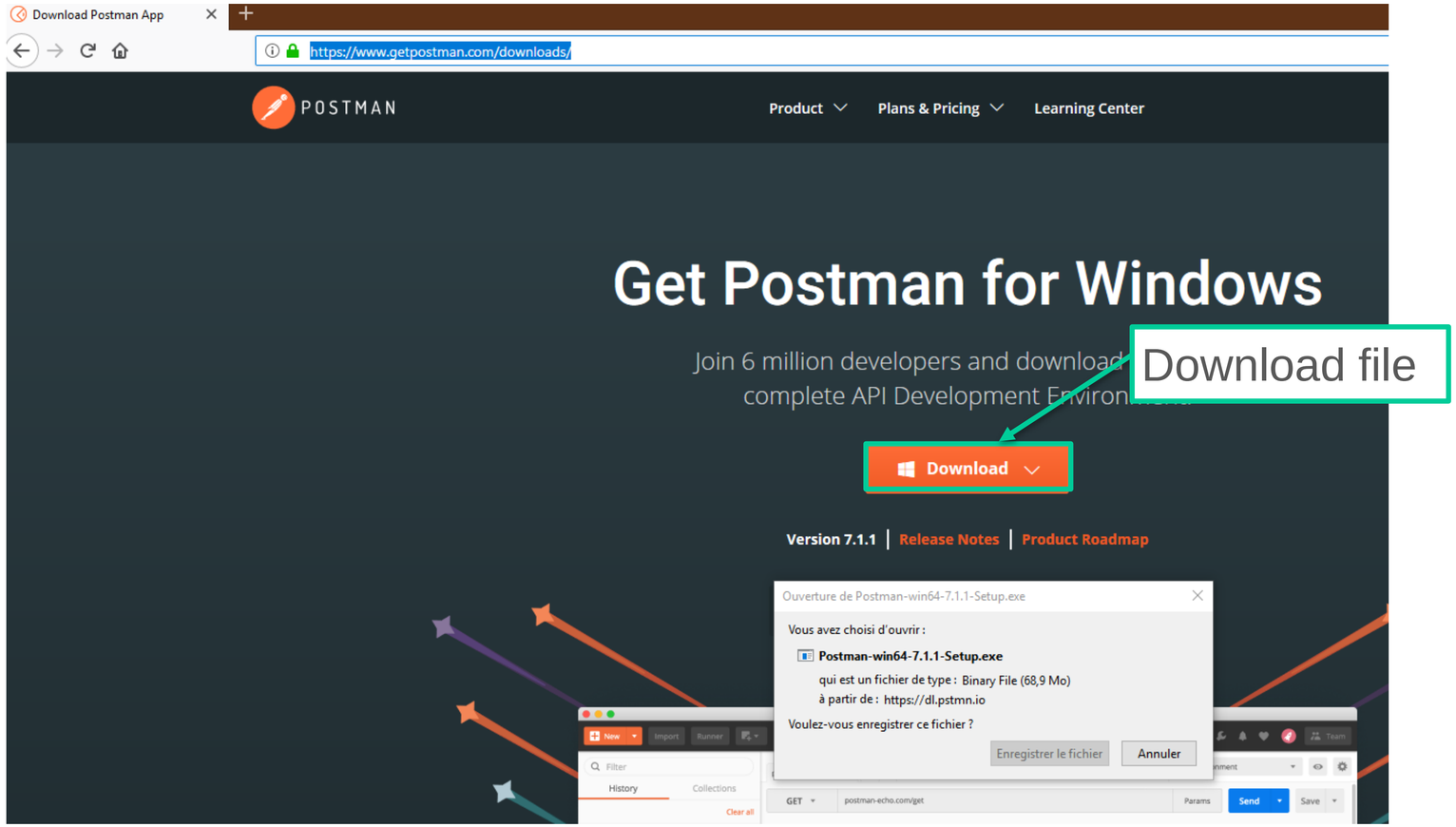
clientETSI DN is **OK**

Accepted

ETSI client

1. Download and Install Postman

- Go to https://www.getpostman.com/downloads/
- Download the file
- Install the tool



ETSI client

2. Get security parameters from encryptors

- Customer has to generate the **client files**
- **Customer uses its CA** (Certificate Authority). It's the **same for both encryptors**
- Get for each encryptors (Alice's and Bob's side)
  - Certificate
  - Key file

This CA file must be given to IDQ

Certificate Authority from customer

Files for Alice

Files for Bob

ca.cert.pem	16/08/2018 14:48	CMS (S/MIME) File	2 Ko
client1-cert.pem	08/03/2019 11:07	CMS (S/MIME) File	3 Ko
client1-key.pem	08/03/2019 11:06	CMS (S/MIME) File	4 Ko
client2-cert.pem	08/03/2019 11:08	CMS (S/MIME) File	3 Ko
client2-key.pem	08/03/2019 11:07	CMS (S/MIME) File	4 Ko

ETSI client

3. Get security parameters from QNCs

- IDQ will provide certificates. They should be loaded into each QNC
- IDQ uses its own CA (Certificate Authority). It's the same for both QNC
- Get for each QNC (Alice and Bob)
  - Certificate
  - Key file

This CA file must be given to the customer

Certificate Authority from IDQ

Files for Alice

Files for Bob

ca.cert.pem	15/02/2019 11:51	CMS (S/MIME) File
qnc-master.cert.pem	26/04/2019 17:18	CMS (S/MIME) File
qnc-master.key.pem	26/04/2019 17:16	CMS (S/MIME) File
qnc-slave.cert.pem	26/04/2019 17:18	CMS (S/MIME) File
qnc-slave.key.pem	26/04/2019 17:16	CMS (S/MIME) File

ETSI client

4. QNC configuration **– Mandatory**

- Put the **client cert CA** files into the root folder of the **USB stick** (formatted in FAT32)
- USB stick must be plugged into the **port number 1 of the QNC**
- Do this procedure for each QNC.
- Import the customer's CA file by connecting to the QNC and use the command encryptor add-client-ca
- Confirm with Y the import
- The **client cert CA** is imported

ETSI client

4. QNC configuration

- In some cases you could import the certificates in the QNC's side

- For this you need to use the commands:
  - encryptor load-key
  - encryptor set-server-keys

ETSI client

4. QNC configuration

- Use the command **encryptor load-key** to load from the USB stick to the QNC the **necessary files.**
- **You need to do it 2 times (cert + key)**
    - **cert file**
    - **key file**

```
Usb drive detected
Usb drive opened
Pem files:
 1) IDQCA.cert.pem.pem
 2) kms-master.pem
 3) kms-master-key.pem
 4) kms-slave.pem
 5) kms-slave-key.pem

Please select an file (between 1 and 5)
2
Key loaded
Usb drive closed
```

ETSI client

4. QNC configuration

- Do it for QNC-master (Alice)

```
Usb drive detected
Usb drive opened
Pem files:
 1) IDQCA.cert.pem.pem
 2) kms-master.pem
 3) kms-master-key.pem
 4) kms-slave.pem
 5) kms-slave-key.pem

Please select an file (between 1 and 5)
2
Key loaded
Usb drive closed
```

```
Usb drive detected
Usb drive opened
Pem files:
 1) IDQCA.cert.pem.pem
 2) kms-master.pem
 3) kms-master-key.pem
 4) kms-slave.pem
 5) kms-slave-key.pem

Please select an file (between 1 and 5)
3
Key loaded
Usb drive closed
```

ETSI client

4. QNC configuration

- Do it for QNC-slave (Bob)

```
Usb drive detected
Usb drive opened
Pem files:
 1) IDQCA.cert.pem.pem
 2) kms-master.pem
 3) kms-master-key.pem
 4) kms-slave.pem
 5) kms-slave-key.pem

Please select an file (between 1 and 5)
4
Key loaded
Usb drive closed
```

```
Usb drive detected
Usb drive opened
Pem files:
 1) IDQCA.cert.pem.pem
 2) kms-master.pem
 3) kms-master-key.pem
 4) kms-slave.pem
 5) kms-slave-key.pem

Please select an file (between 1 and 5)
5
Key loaded
Usb drive closed
```

ETSI client

4. QNC configuration

- Use the command **encryptor set-server-keys** to configure the **cert and key files** to the specified encryptor created on KEMS (previously)

```
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]
```

ETSI client

4. QNC configuration

- Do it for QNC-SLAVE

```
admin> encryptor set-server-keys
         NAME      CERT-FILE            KEY-FILE           EXPIRATION                                          SUBJECT
        ETSI2   kms-cert.pem         kms-key.pem  02/10/2023 09:05:00  CN=KME1, OU=QuantumSafe, O=ID Quantique, L=Gene...
   centaurisB  QKDServer.pem  QKDServer_pkey.pem  09/11/2035 18:42:05  CN=QKDServer, OU=QuantumSafe, O=ID Quantique, L...
Please type the consumer name for which you wish to change the server keys.
SYSTEM NAME:  centaurisB
CERT FILE:    QKDServer.pem
KEY FILE:      QKDServer_pkey.pem
Are you sure you want to continue? [Y/n]
```

ETSI client

4. QNC configuration

- Check the using files with the “encryptor” command

```
admin> encryptor
         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...
==================================
TRUSTED CAs
----------------------------------
OU=Security, O=ID Quantique, L=Geneva, S=Geneva State, C=CH
OU=Test, S=MA, userId=1599848659, CN=CommonName, O=Test, C=US
----------------------------------
admin>
```

ETSI client

4. QNC configuration

- Do it on QNC-SLAVE

```
admin> encryptor
         NAME      CERT-FILE            KEY-FILE           EXPIRATION                                                SUBJECT
        ETSI2   kms-cert.pem         kms-key.pem  02/10/2023 09:05:00  CN=KME1, OU=QuantumSafe, O=ID Quantique, L=Gene...
   centaurisB  QKDServer.pem  QKDServer_pkey.pem  09/11/2035 18:42:05  CN=QKDServer, OU=QuantumSafe, O=ID Quantique, L...
==================================
TRUSTED CAs
----------------------------------
OU=Security, O=ID Quantique, L=Geneva, S=Geneva State, C=CH
OU=Test, S=MA, userId=1599848659, CN=CommonName, O=Test, C=US
----------------------------------
admin>
```

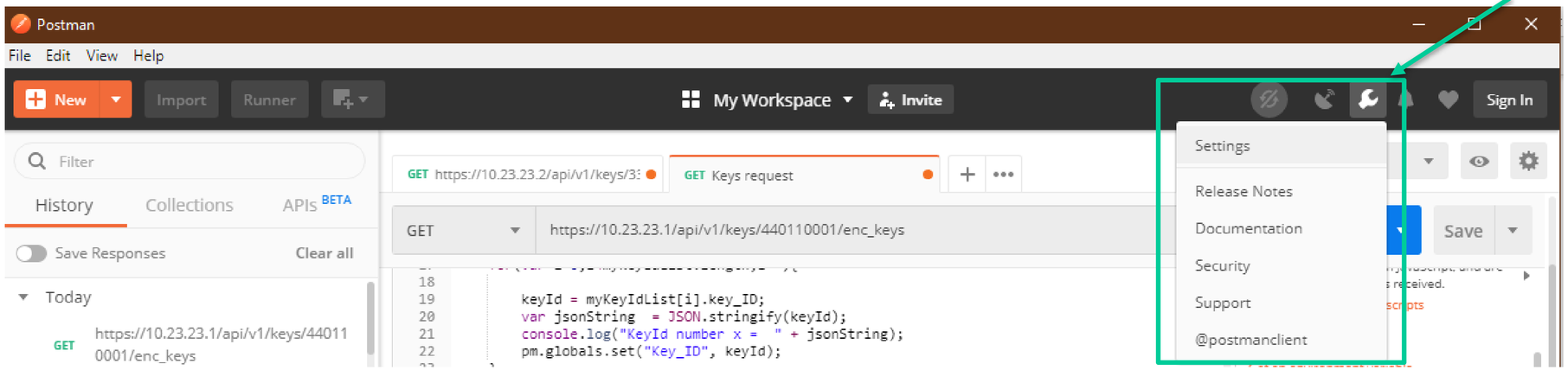
Note: You may use separate certificates/keys for the master vs slave QNC.

SWISS QUANTUM

5. Add CA file on Postman

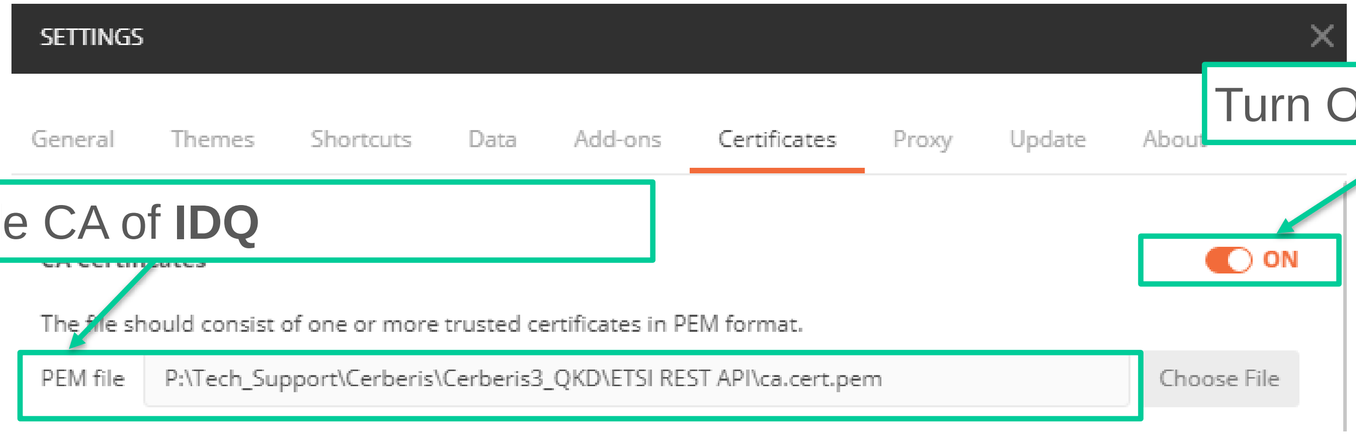
- Add the Certification Authority

Go to Settings



Turn ON CA

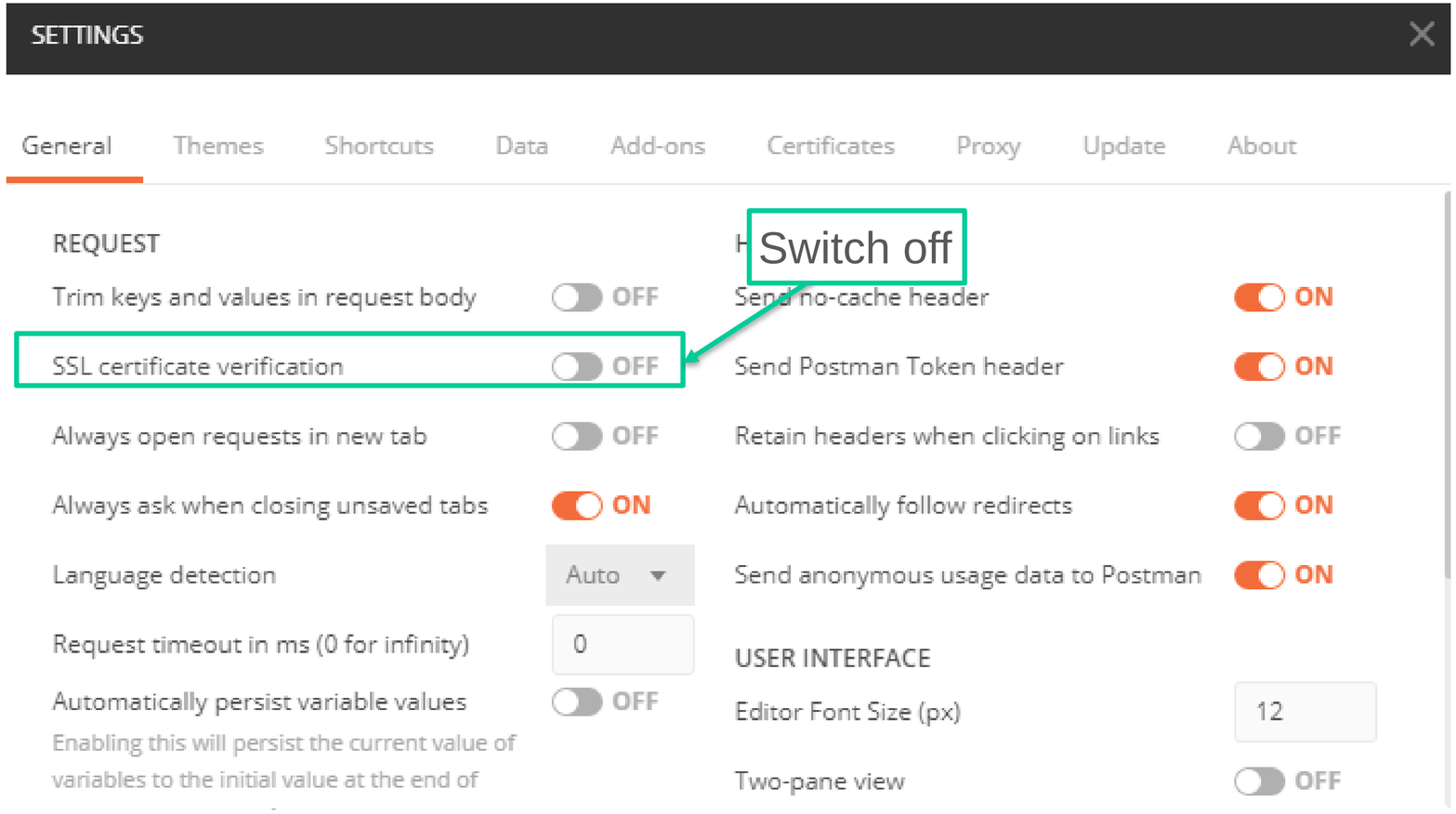
Add the .pem file CA of **IDQ**



IDQ

SWISS QUANTUM

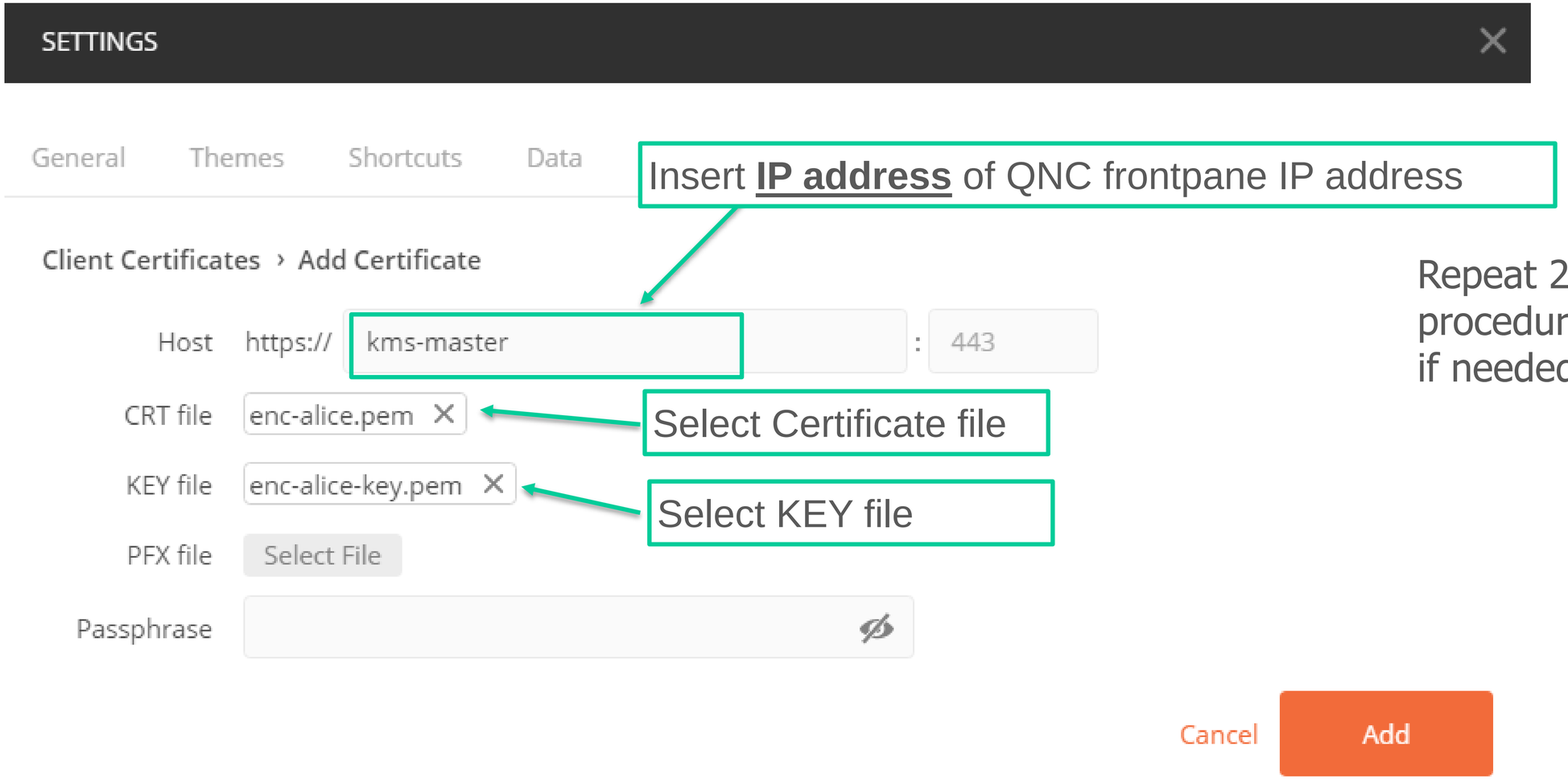
6. Disable the SSL certificate verification (because **CA is not signed**)



IDQ

SWISS QUANTUM

7. Add client key file and cert files



Insert **IP address** of QNC frontpane IP address

Select Certificate file

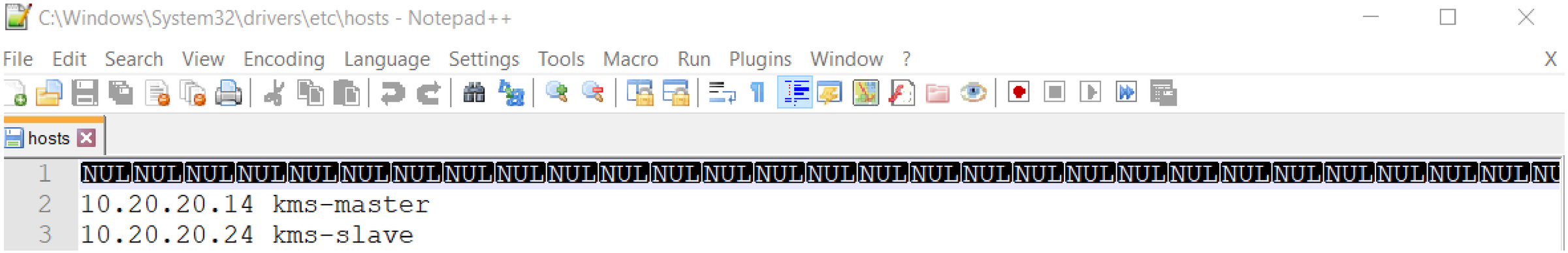
Select KEY file

Repeat 2 times this procedure: for each encryptor if needed

IDQ

8. Modify your host files (optional)

- Indicate on your host file the IP addresses and hostname of the frontpane
- The host file is localized in a different place depending of the OS



SWISS QUANTUM

9. Check if you can reach front pane by host name

- You should be able to ping both front pane by host names

Command Prompt

```
Microsoft Windows [Version 10.0.18362.657]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\arnaud.camandona>ping kms-master

Pinging kms-master [10.20.20.14] with 32 bytes of data:
Reply from 10.20.20.14: bytes=32 time<1ms TTL=64

Ping statistics for 10.20.20.14:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
Control-C
^C
C:\Users\arnaud.camandona>ping kms-slave

Pinging kms-slave [10.20.20.24] with 32 bytes of data:
Reply from 10.20.20.24: bytes=32 time<1ms TTL=64
Reply from 10.20.20.24: bytes=32 time<1ms TTL=64

Ping statistics for 10.20.20.24:
    Packets: Sent = 2, Received = 2, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
Control-C
^C
C:\Users\arnaud.camandona>
```


IDQ

10. DN adding

- **Find and copy** the **DN** of **each client** certificate using OPENSSL (subject line)
- **Find and copy** the **CN** of **each client** certificate using OPENSSL (subject line)

```
C:\Program Files\OpenSSL-Win64\bin>openssl.exe x509 -in "C:\Users\Public\Documents\Cert\Files\enc-alice.pem" -noout -text
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            75:0d:8c:1a:15:3a:04:35:b8:c8:19:06:1f:a2:78:76:94:06:5a:a8
        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-alice
        Subject Public Key Info:
            Public Key Algorithm: id-ecPublicKey
                Public-Key: (256 bit)
                pub:
                    04:75:8b:d4:b2:31:73:ac:7a:2d:cd:80:52:29:0a:
                    ee:3d:43:71:22:c7:cf:8b:b3:d4:23:65:6a:a4:52:
                    62:57:29:b1:8e:45:d1:9e:a3:86:e6:ed:76:3f:bb:
                    11:4c:88:47:51:5c:d8:04:1c:ae:6b:7a:3d:36:c0:
                    14:b6:5b:e4:49
                ASN1 OID: prime256v1
                NIST CURVE: P-256
        X509v3 extensions:
```

CN

DN

SWISS QUANTUM

10. DN adding

- **Do it for client slave**

```
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:
```

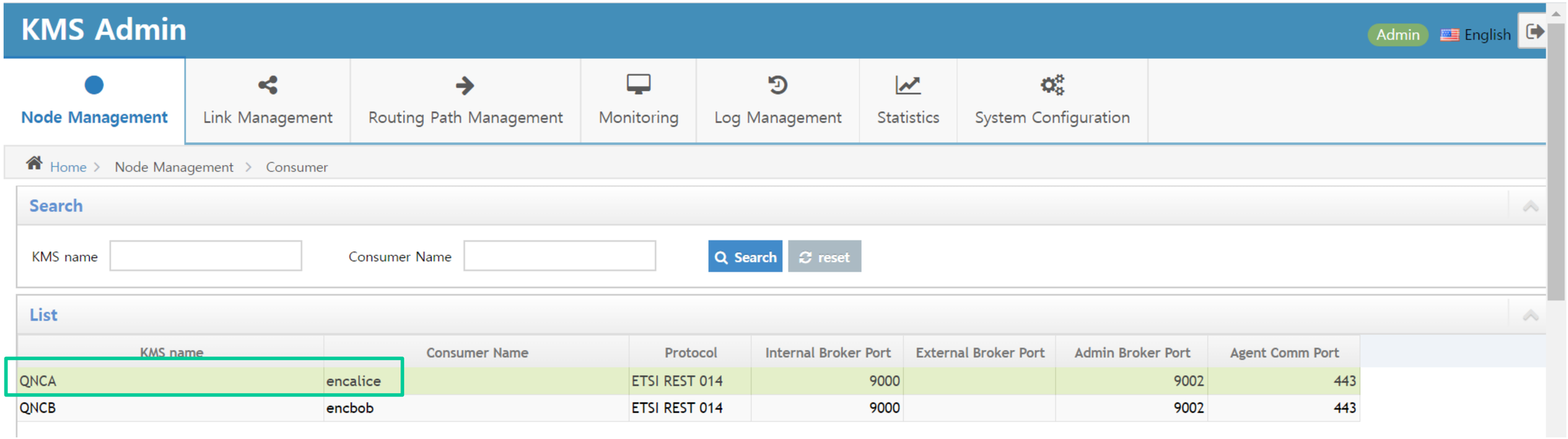
CN

DN

IDQ

10. DN adding

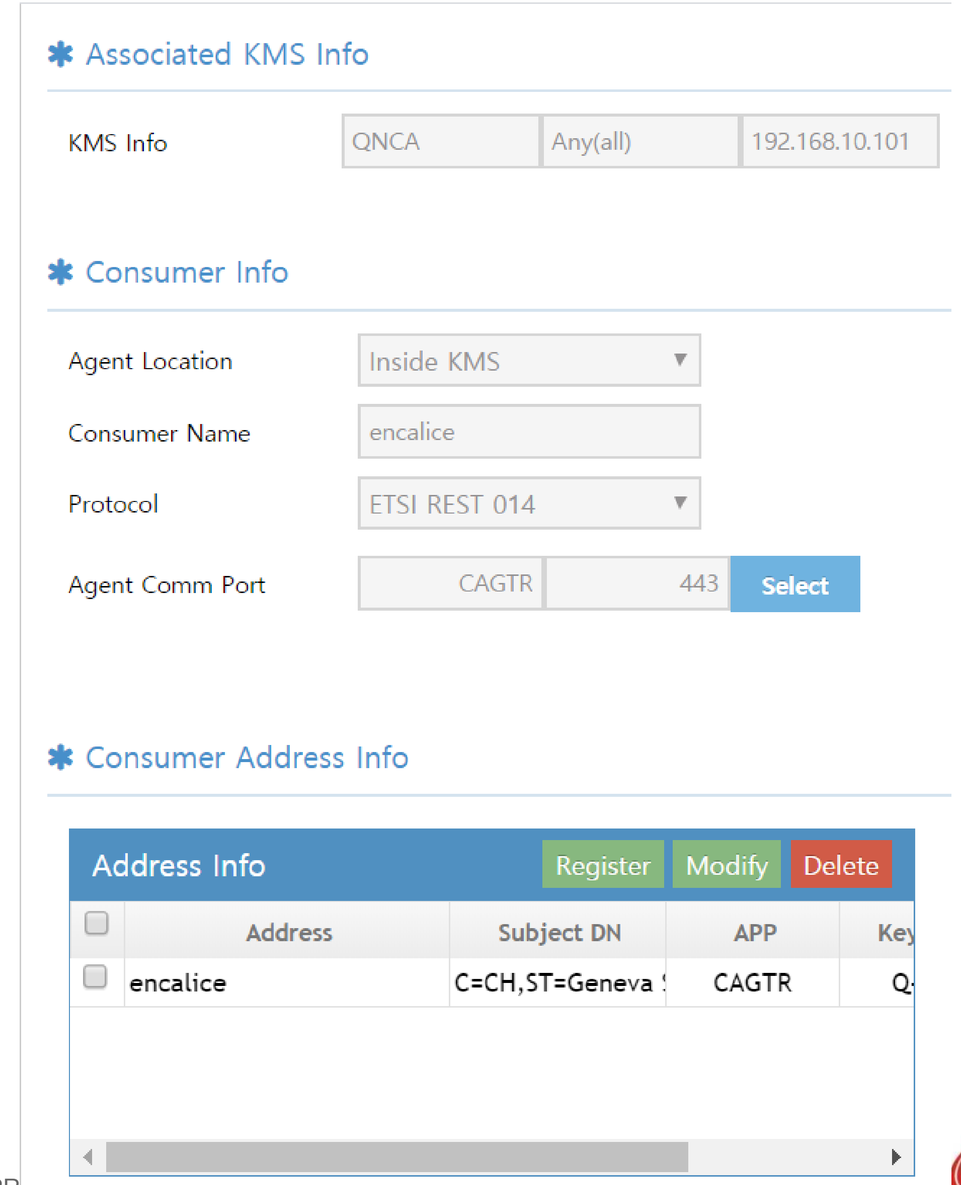
- Go on KEMS -> Consumer
- Localize and select your encryptor
- You may need to create a new one if you were using different encryptors



ETSI client

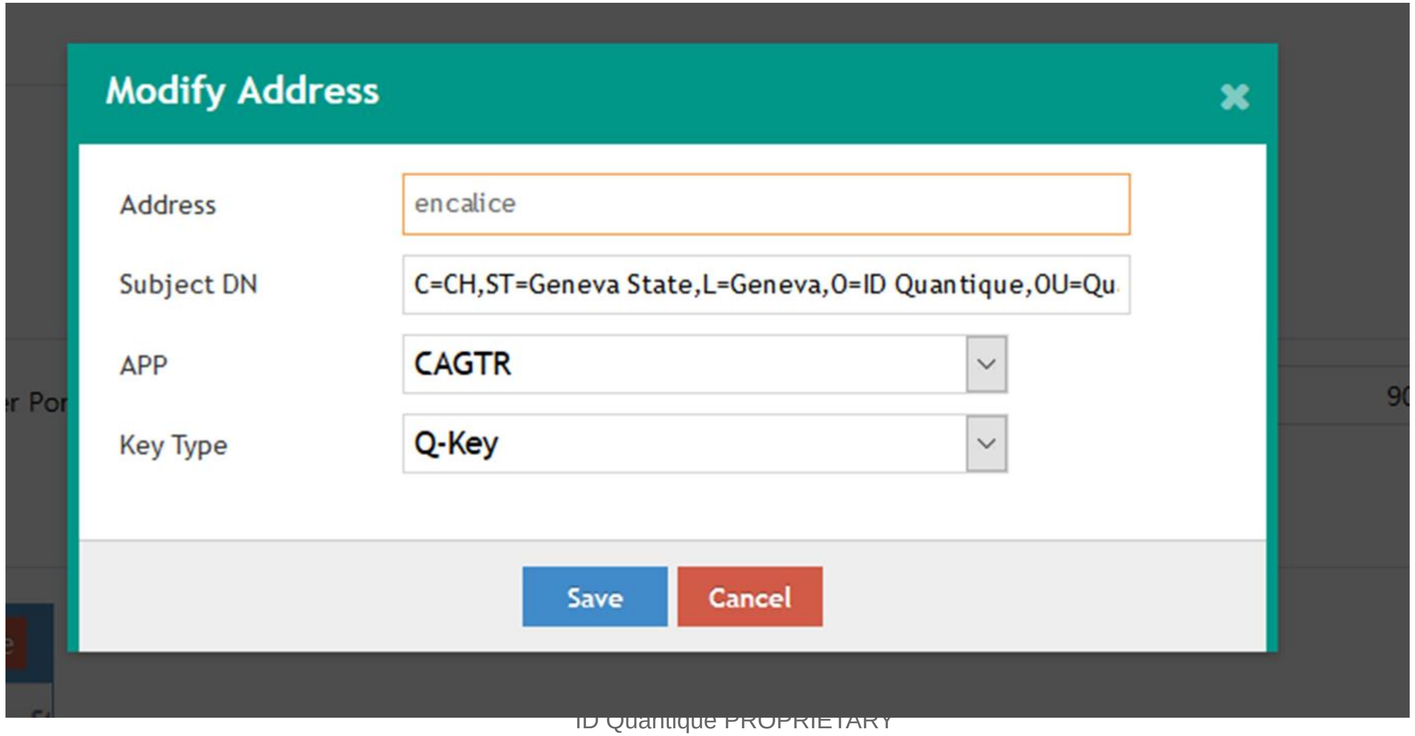
10. DN adding

- Scroll down
- For ETSI here are the parameters:
  - Inside KMS (for ETSI)
  - Name of Consumer
  - Select Protocol (ETSI REST 014)
  - Agent Comm Port = CAGTR/443 (for ETSI)



10. Adding Encryptor names

- Create or modify your encryptor and insert the name of your client. This name will be used on the command to get keys
- This name needs to be short and not contain space or symbols. Select CAGTR and Q-KEY
- Copy paste the Subject DN (that you copied from OpenSSL) without spaces and save
- Save one time more on the previous main page



ID Quantique PROPRIETARY

10. DN adding

- Spelling of Subject DN is quite important.
- No space is allowed, if OpenSSL gives you a Subject DN with spaces, you need to remove these spaces

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

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

SubjectDN must follow the RFC-4514 standard format and must be in same order as given in the encryptor side certificate (aka the client certificate). So, use comma and no spaces to separate fields. Notice that some Certificate tools are not correct and don't display this SubjectDN correctly (OpenSSL sometimes display fields using slash (/) or spaces as separators, even ADVA also don't follow this standard… **so triple check!**

ETSI tests

1. Test if the key status command is working on Alice

GET https://kms-master/api/v1/k... | GET https://kms-master/api/v1/k...

No Env

Insert the command

GET

https://**IP_ADDRESS_QNCALICE**/api/v1/keys/**encbob**/status

Send

Save

Pretty | Raw | Preview | Visualize | Text

Send command


```
1  {
2    "source_application_id":"V3000000100001",
3    "target_application_id":"V3000000200001",
4    "source_id":"V2000000300001",
5    "target_id":"V2000000200001",
6    "key_size":256,
7    "stored_key_count":91,
8    "max_key_count":100,
9    "max_key_per_request":1,
10   "max_key_size":256,
11   "min_key_size":128
12 }
```


System gives you a status

SWISS QUANTUM

2. Test if the key status command is working on Bob

GET https://kms-master/api/v1/k... GET https://kms-master/api/v1/k...

No Env

Insert the command

GET https://**IP_ADDRESS_QNCBOB**/api/v1/keys/**encalice**/status

Send Save

Send command

Pretty Raw Preview Visualize Text

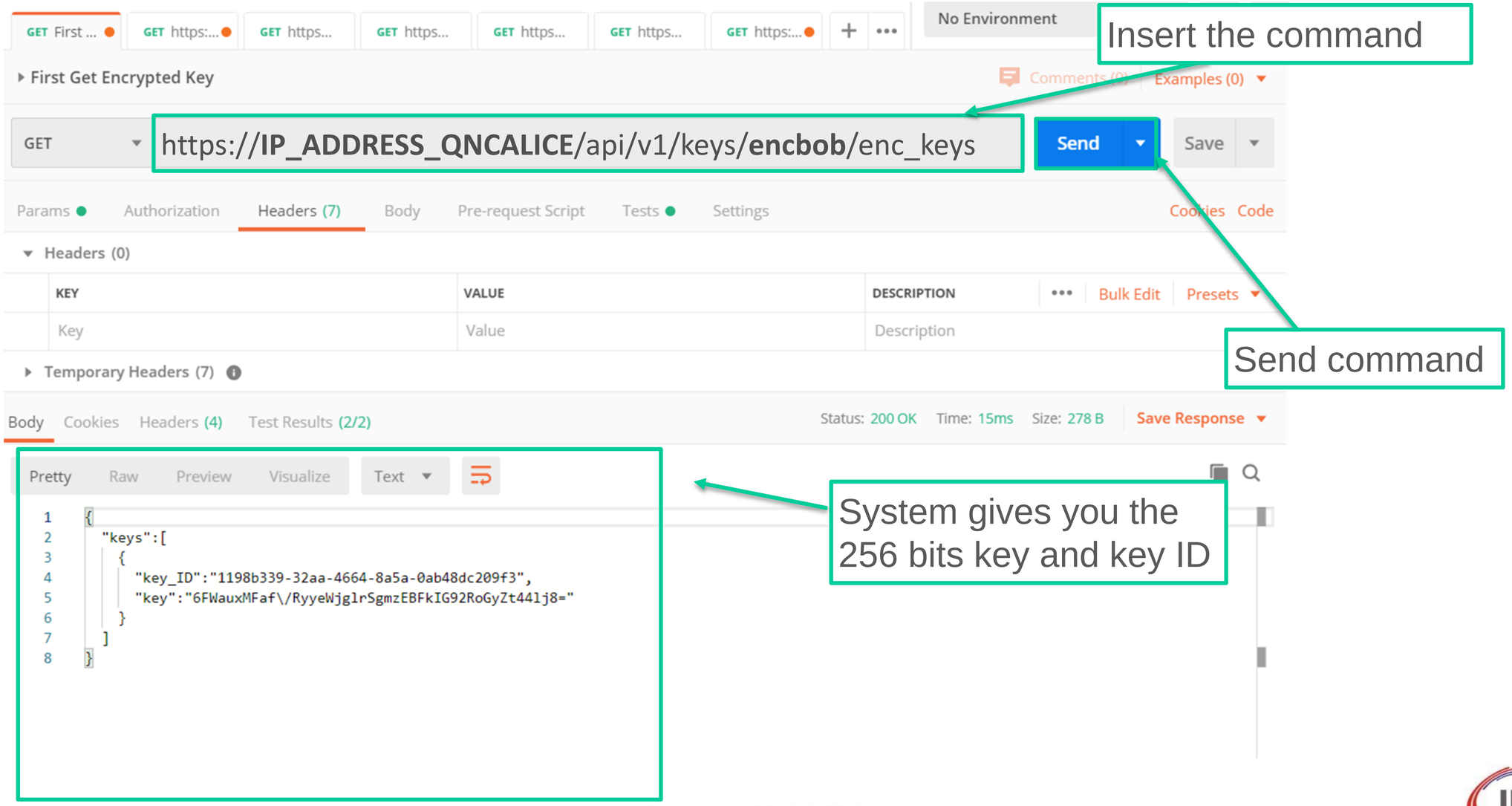

```
{
  "source_application_id":"V3000000100001",
  "target_application_id":"V3000000200001",
  "source_id":"V2000000300001",
  "target_id":"V2000000200001",
  "key_size":256,
  "stored_key_count":91,
  "max_key_count":100,
  "max_key_per_request":1,
  "max_key_size":256,
  "min_key_size":128
}
```


System gives you a status

IDQ

SWISS QUANTUM

3. Test if the key get command is working



Insert the command

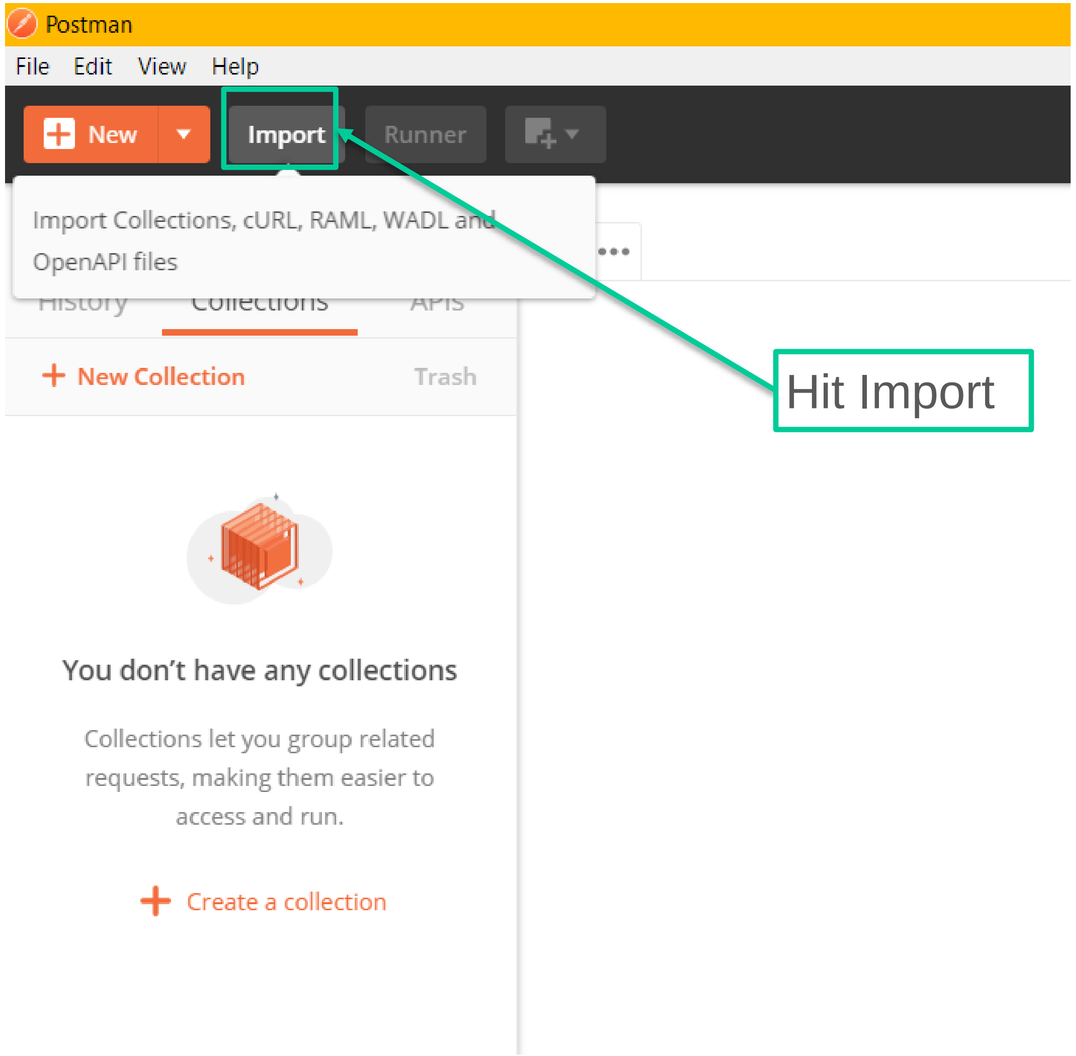
Send command

System gives you the 256 bits key and key ID

IDQ

SWISS QUANTUM

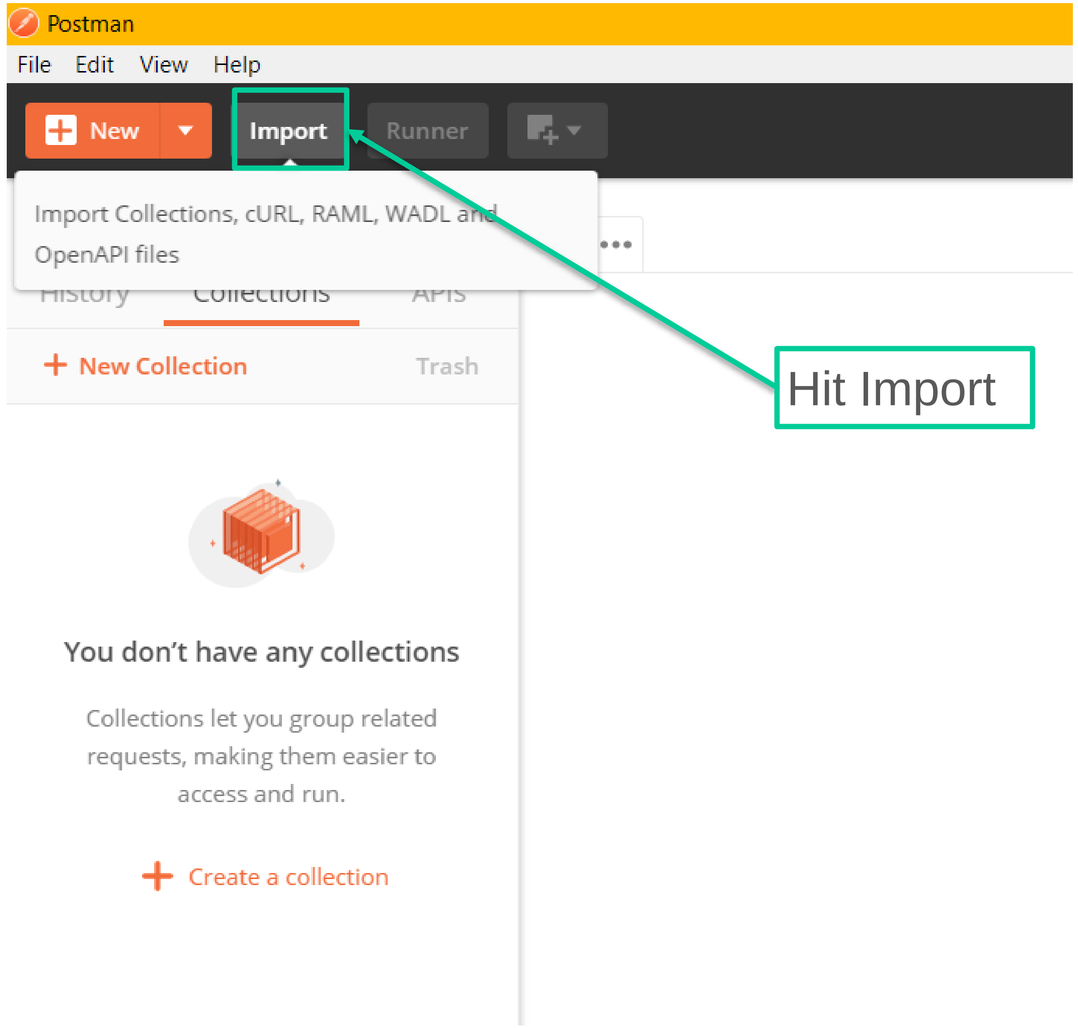
4. Test simultaneously the encrypted key on both QNC

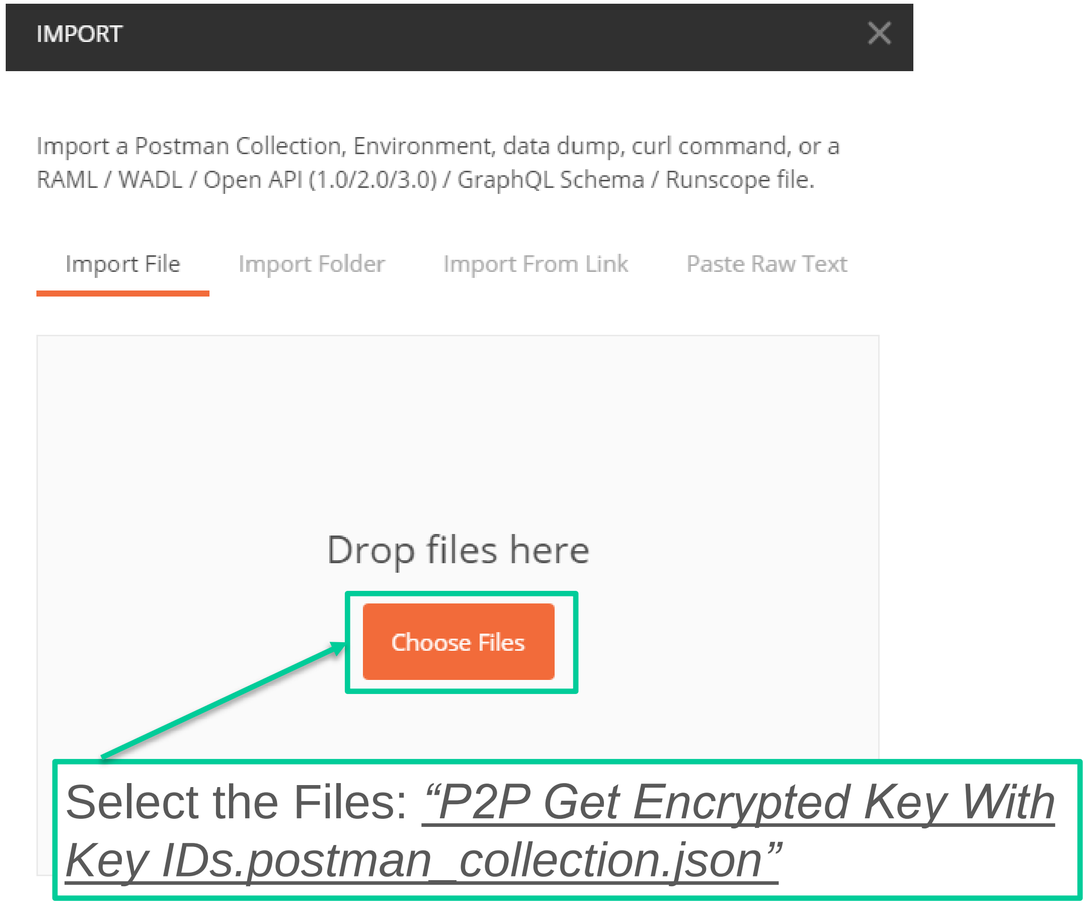


IDQ

SWISS QUANTUM

4. Test simultaneously the encrypted key on both QNC



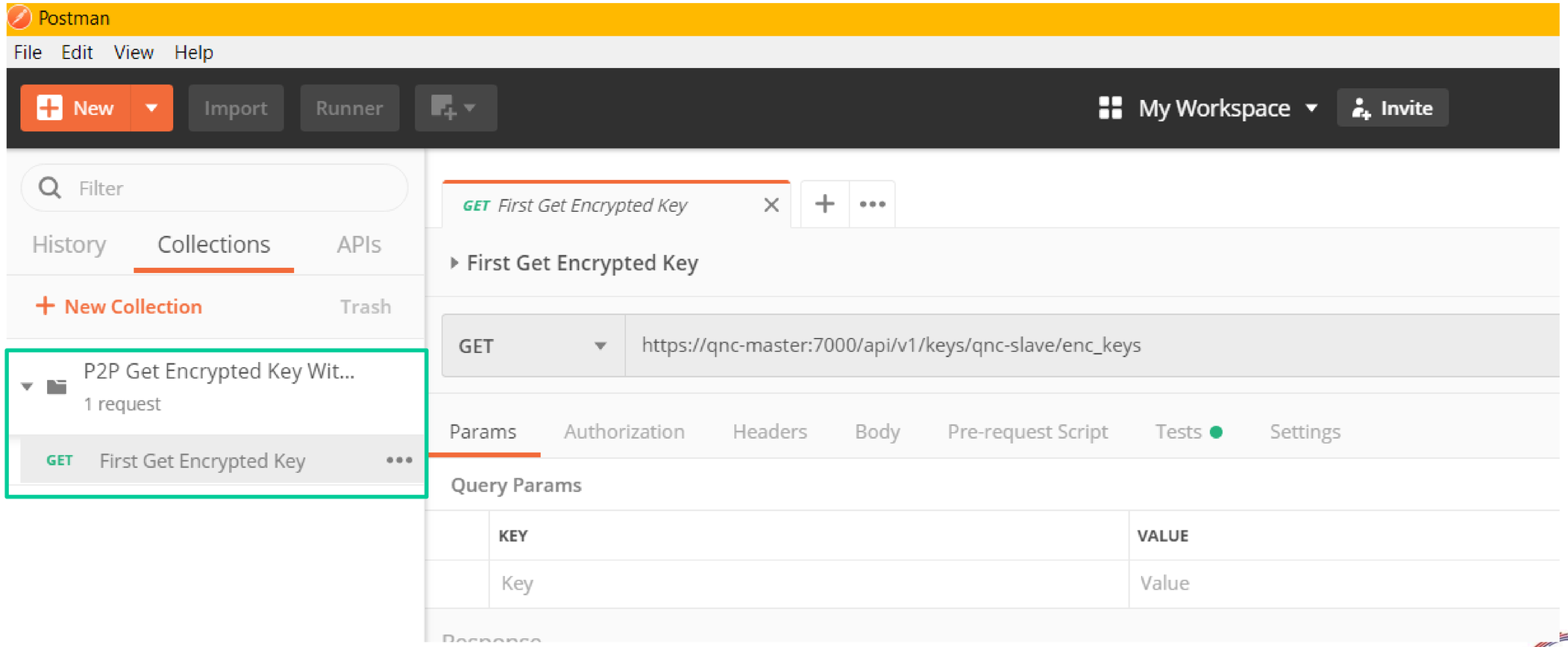


IDQ

SWISS QUANTUM

4. Test simultaneously the encrypted key on both QNC

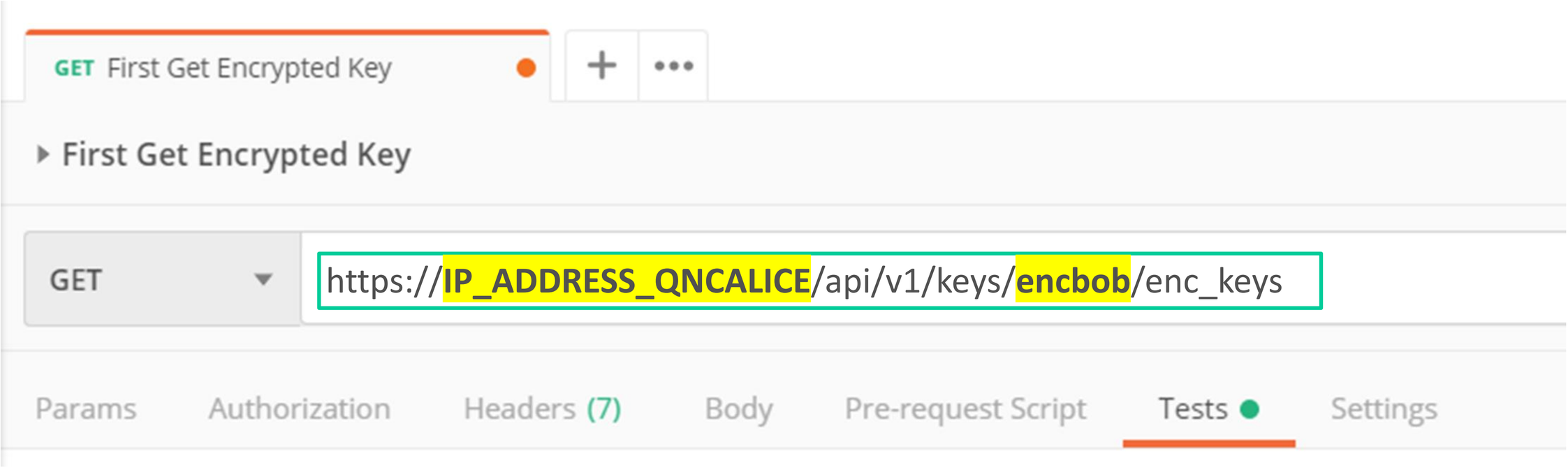
- The script should appear at the left side, select ID
- Expand it and select the GET line



IDQ

4. Test simultaneously the encrypted key on both QNC

- Edit the get line for “enc_keys”. You need to change all of these values:
  - kms-master which is the QNC master
  - enc-bob which is the encryptor in the Bob (slave) side
  - Port 443 as it was configured on the consumer part of KEMS



SWISS QUANTUM

4. Test simultaneously the encrypted key on both QNC

- Go to the tests tab

GET First Get Encrypted Key

First Get Encrypted Key

GET https://IP_ADDRESS_QNCALICE/api/v1/keys/encbob/dec_keys

Params Authorization Headers Body Pre-request Script Tests Settings

```
        console.log("KeyId number x =  " + jsonString);
        pm.globals.set("Key_ID", keyId);
    }
});


// Now Try to retrieve the key by his Id from the slave
const SendGetKeyByIds = {
  url: 'https://qnc-slave:7000/api/v1/keys/qnc-master/dec_keys',
  method: 'POST',
  header: 'headername1:value1',
  body: {
    mode: 'raw',
    raw: JSON.stringify({key_IDs: [{key_ID: pm.globals.get("Key_ID") }]})
  }
};

pm.sendRequest(SendGetKeyByIds, function (err, res) {
  console.log(err ? err : res.json());
});

```


IDQ

SWISS QUANTUM

4. Test simultaneously the encrypted key on both QNC

- Modify the following information, QNC name, encryptor and port:

Params | Authorization | Headers | Body | Pre-request Script | Tests | Settings

```
            console.log("KeyId number x =  " + jsonString);
            pm.globals.set("Key_ID", keyId);
        }
});


// Now Try to retrieve the key by his Id from the slave
const SendGetKeyByIds = {
  url:'https://IP_ADDRESS_QNCBOB/api/v1/keys/encalice/dec_keys'
  method: 'POST',
  header: 'headername1:value1',
  body: {
    mode: 'raw',
    raw: JSON.stringify({key_IDs: [{key_ID: pm.globals.get("Key_ID") }]})
  }
};

pm.sendRequest(SendGetKeyByIds, function (err, res) {
  console.log(err ? err : res.json());
});

```


IDQ

SWISS QUANTUM

4. Test simultaneously the encrypted key on both QNC

- Hit SEND

GET First Get Encrypted Key

No Environment

First Get Encrypted Key

Comments 0

GET https://10.10.20.240:443/api/v1/keys/encbob/enc_keys

Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings

```
        keyId = myKeyIdList[i].key_ID;
        var jsonString  = JSON.stringify(keyId);
        console.log("KeyId number x =  " + jsonString);
        pm.globals.set("Key_ID", keyId);
    }
});


// Now Try to retrieve the key by his Id from the slave
const SendGetKeyByIds = {
  url: 'https://10.10.20.247:443/api/v1/keys/encalice/dec_keys',
  method: 'POST',
  header: 'headername1:value1',
  body: {
    mode: 'raw',
    raw: JSON.stringify({key_IDs: [{key_ID: pm.globals.get("Key_ID") }]})
  }
};

pm.sendRequest(SendGetKeyByIds, function (err, res) {
  console.log(err ? err : res.json());
});
```

Test scripts are written in JavaSc
run after the response is receive
Learn more about tests scripts

SNIPPETS

Get an environment variable

Get a global variable

Get a variable

Set an environment variable

Set a global variable

Clear an environment variable

Clear a global variable

IDQ

ETSI tests

4. Test simultaneously the encrypted key on both QNC

- The system gives you a key which is the same for both sides with no error (Test Results 2/2)

Body | Cookies | Headers (4) | Test Results (2/2)

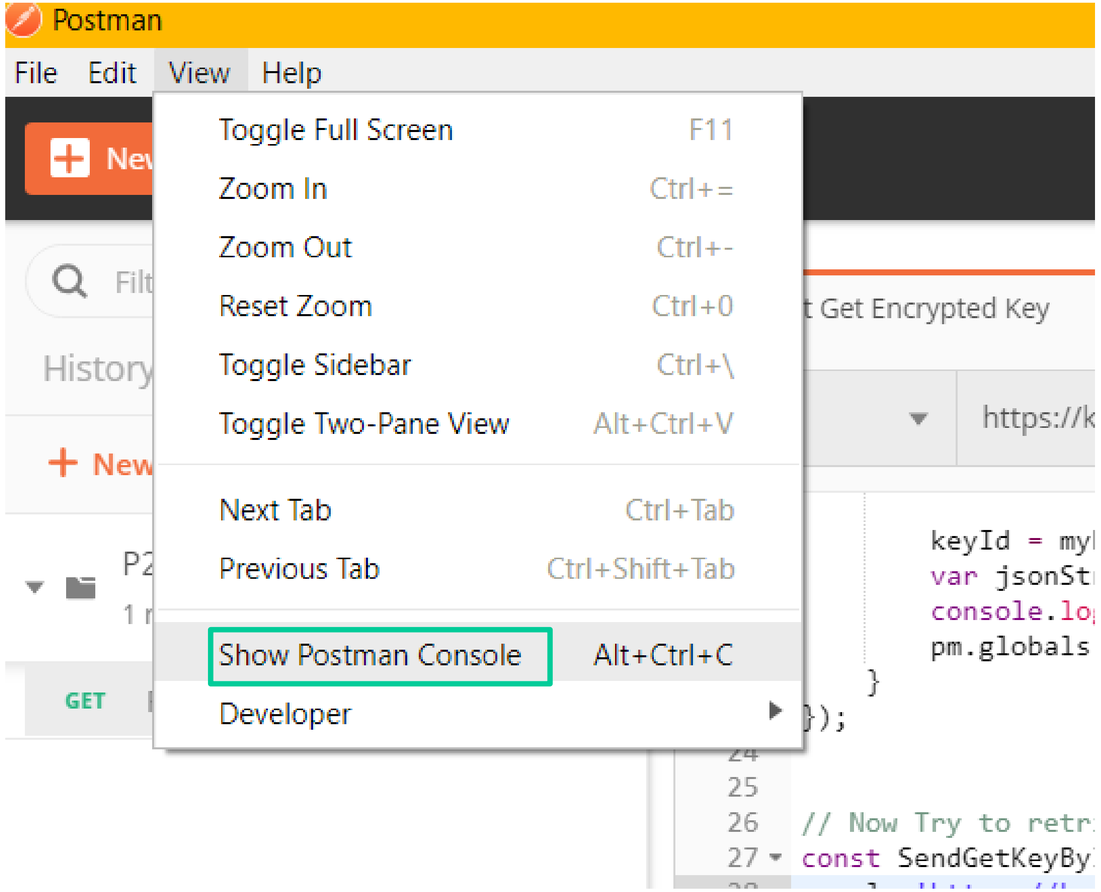
Pretty | Raw | Preview | Visualize | Text


```
{
  "keys":[
    {
      "key_ID":"ed452d45-0aa2-47ea-a5be-165827b1ea4f",
      "key":"dVzTzttpEN6swak+YmXNoy5om9w7mbhmp0v2pWns5U0="
    }
  ]
}
```

SWISS QUANTUM

4. Test simultaneously the encrypted key on both QNC

- Open Postman Console

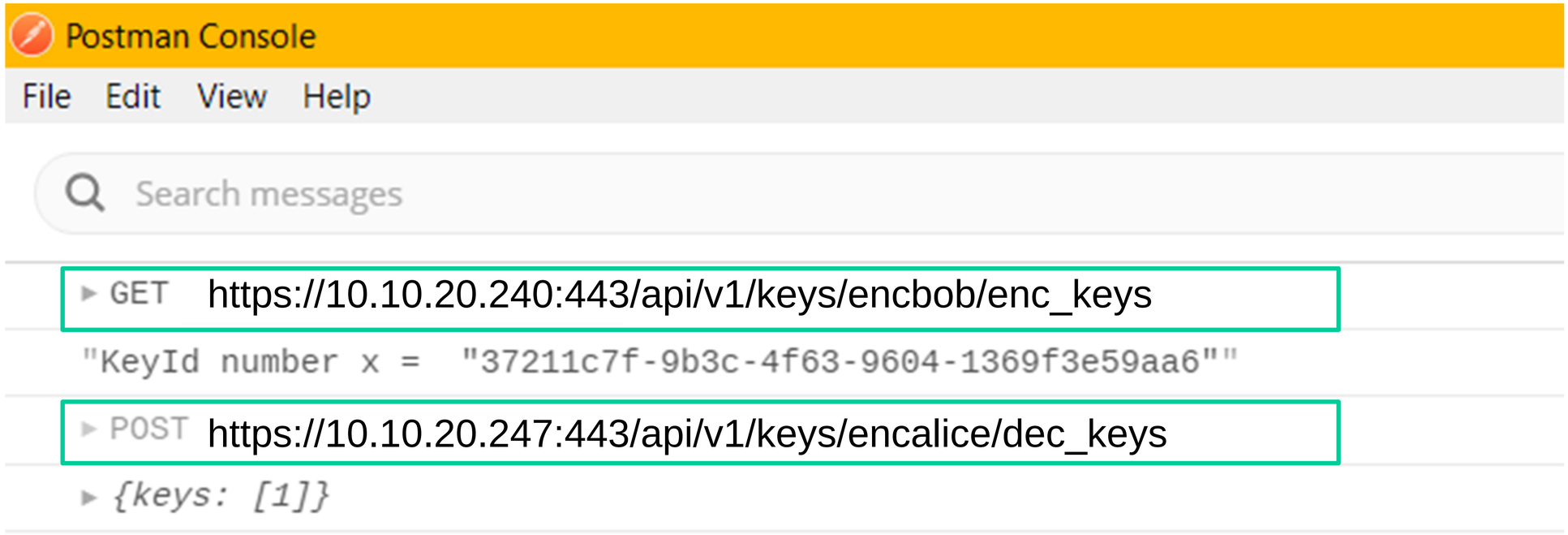


IDQ

ETSI tests

4. Test simultaneously the encrypted key on both QNC

- Open Postman Console
- You can see the POST command and the GET command



ETSI tests

4. Test simultaneously the encrypted key on both QNC

- If you expand GET you can see the key and key ID

Postman Console

File Edit View Help

Search messages

```
▾ GET https://10.10.20.240:443/api/v1/keys/encbob/enc_keys
  ▸ Client Certificate
  ▸ Network
  ▾ Request Headers
    User-Agent: "PostmanRuntime/7.22.0"
    Accept: "*/*"
    Cache-Control: "no-cache"
    Postman-Token: "e1aa4755-7e35-4156-8637-3c2125ee7646"
    Host: "kms-master:443"
    Accept-Encoding: "gzip, deflate, br"
    Connection: "keep-alive"
  ▾ Response Headers
    Connection: "Keep-Alive"
    Content-Length: "144"
    Access-Control-Allow-Origin: "*"
    Date: "Wed, 04 Mar 2020 16:02:44 GMT"
  ▾ Response Body
    ▾ keys: [1]
      ▾ 0: {…}
        key_ID: "37211c7f-9b3c-4f63-9604-1369f3e59aa6"
        key: "vIw2IprOgc2WpLYfbFF35UKTXvz1bPhegsAk+JdV6so="
"KeyId number x =  "37211c7f-9b3c-4f63-9604-1369f3e59aa6""
▸ POST https://10.10.20.247:443/api/v1/keys/encalice/dec_keys
▸ {keys: [1]}
```

ETSI tests

4. Test simultaneously the encrypted key on both QNC

- If you expand POST you can see the key and key ID

Postman Console

File Edit View Help

Search messages

```
▸ GET https://10.10.20.240:443/api/v1/keys/encbob/enc_keys
"KeyId number x =  "37211c7f-9b3c-4f63-9604-1369f3e59aa6""
▾ POST https://10.10.20.247:443/api/v1/keys/encalice/dec_keys
  ▸ Client Certificate
  ▸ Network
  ▾ Request Headers
    headername1: "value1"
    Content-Type: "text/plain"
    User-Agent: "PostmanRuntime/7.22.0"
    Accept: "*/*"
    Cache-Control: "no-cache"
    Postman-Token: "8ce3846e-36a8-49b8-95ff-61540f54fc9e"
    Host: "kms-slave:443"
    Accept-Encoding: "gzip, deflate, br"
    Content-Length: 63
    Connection: "keep-alive"
  ▸ Request Body
  ▾ Response Headers
    Connection: "Keep-Alive"
    Content-Length: "144"
    Access-Control-Allow-Origin: "*"
    Date: "Wed, 04 Mar 2020 16:02:43 GMT"
  ▾ Response Body
   ▾ keys: [1]
    ▾ 0: {…}
      key_ID: "37211c7f-9b3c-4f63-9604-1369f3e59aa6"
      key: "vIw2IprOgc2WpLYfbFF35UKTXvz1bPhegsAk+JdV6so="
```

ETSI tests

4. Test simultaneously the encrypted key on both QNC

- The values are the **same on both sides**
- Based on these different tests we can say the **QKD is fully working**
- You can use another ETSI client if you want to conduct other tests

Alternatively use this simple shell script instead

```
#!/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 QNC/Frontpane Address, SAE Address, Cert and Key Locations

Output from shell script

Verified/Tested using Ubuntu20

```
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
```

Key ID

Quantum Key

THANKS