



# Quantum Key Distribution

## Training CerberisXG

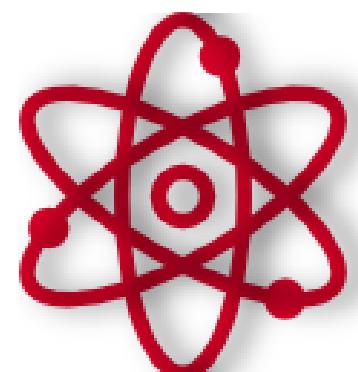
Version 06-September 2022  
For Q2-2022 (V3.2.0 version)

# Company presentation



| Founded in 2001  
| Geneva, Switzerland  
Seoul, South Korea  
Boston, USA  
Vienna, Austria

| By 4 quantum physicists from the University of Geneva  
| ~100 employees in CH, including 50 engineers/scientists  
 Investments in 2018 by SK Telecom &  
 Deutsche Telekom



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



| Performs R&D, production, professional services, integration, support  
| Clients: Governments / Banks / Gaming Industry / Universities / IT Security

## Learning about QKD systems – What you must know

### Goal of this section:

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



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

## Key Markets



Telecom and Data Center Service Providers



Financial Services Companies



Governments and Defence



Healthcare Organizations



Critical Infrastructure



IP-rich Enterprises

## Key Applications



Data center interconnections



Metropolitan backbone optical networks



Long distance distribution using relay nodes



Key distribution across a complex network (ring, hub and spoke, meshed)



Crypto keys as-a-service



Validation of QKD and encryption pilot networks

# CerberisXG presentation

SWISS QUANTUM

## Cerberis XG QKD System at a glance

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



<sup>1</sup> 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>

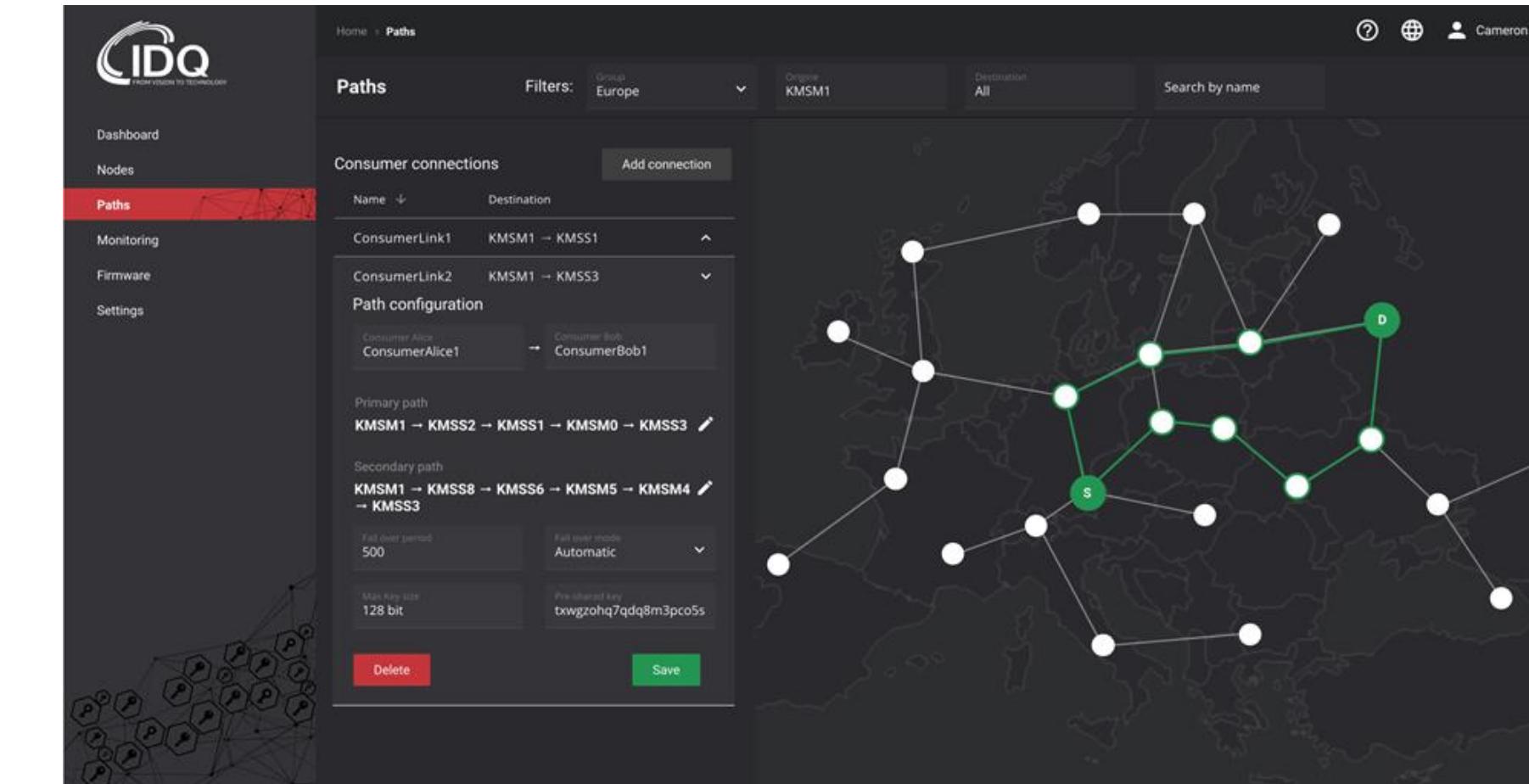
<sup>2</sup> For a typical installation with the ATCA chassis, the QNC and one QKD blade

# At a Glance

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

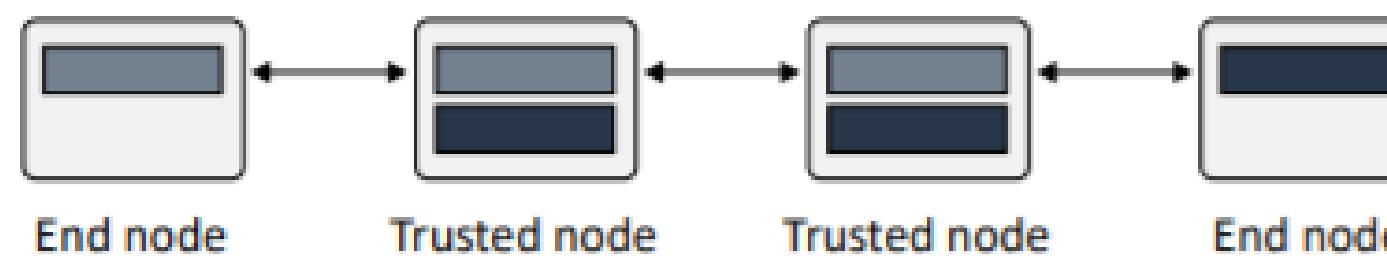


- Embedded KMS and management tools
- Enhanced monitoring (Syslog, SNMP, QMS ...)
- New centralized EMS User Interface – **QMS**

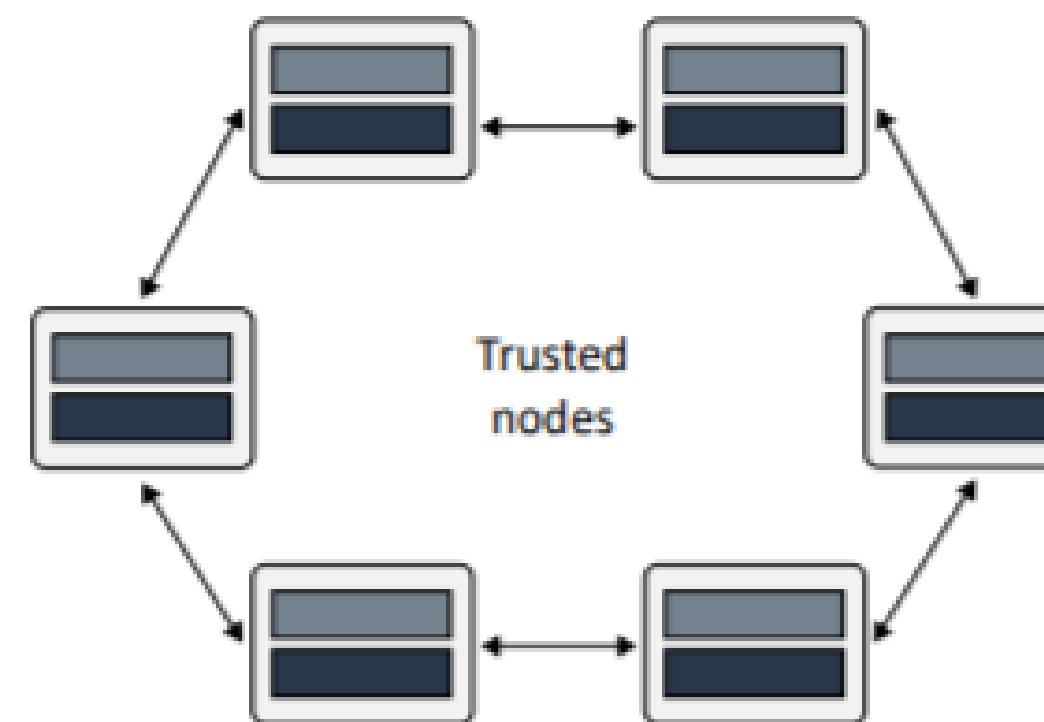


# Supported Topologies

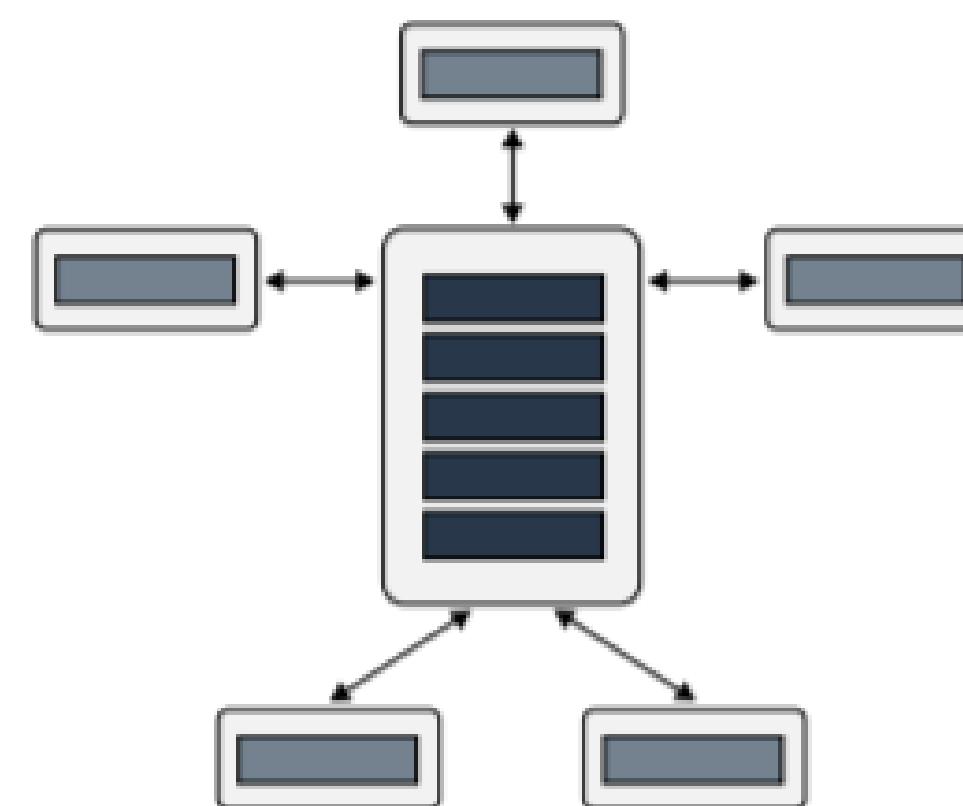
**Point-to-point (with relay for long distance)**



**Ring network**



**Star**



Optical unit Alice - 1U  
 Optical unit Bob - 1U  
 Quantum channel (dark fiber or wavelength in O-band)  
↔ { KMS channel (logical mux possible) / ETH  
Service channel (C-band)

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

- The purpose of the QKD system is quite simple:
  1. Exchange quantum bits on a quantum channel used to create keys
  2. Deliver keys to third party devices (Encryptors or HSM)
  3. Third party devices use these keys to encrypt data

## Network Architecture



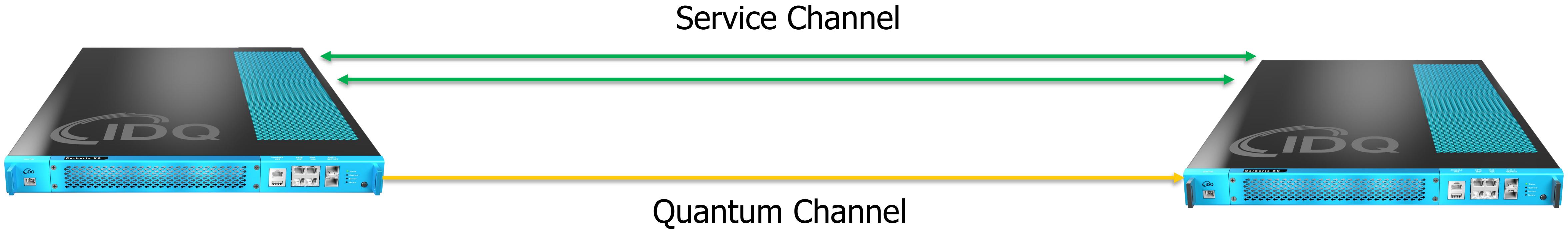
# QKD Topology



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

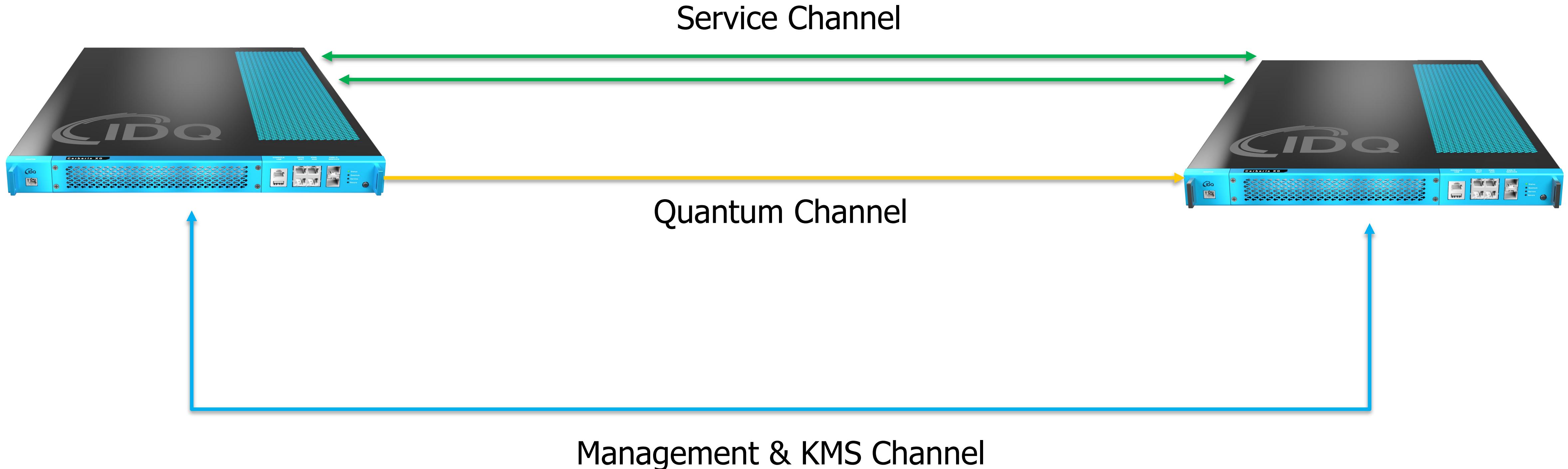
# QKD Topology

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



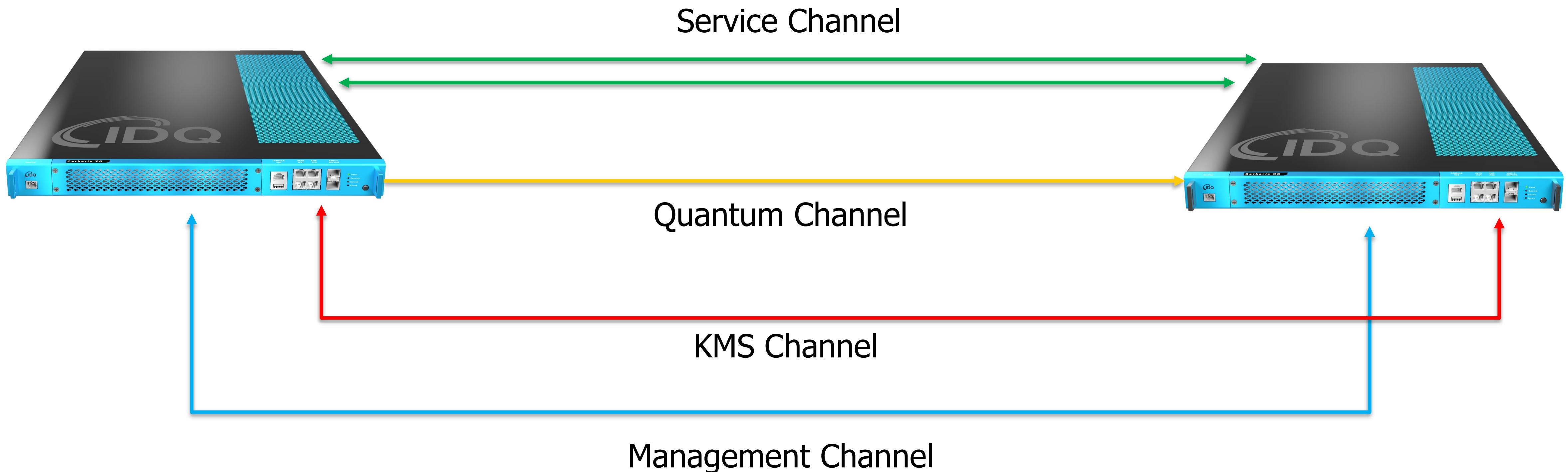
# QKD Topology

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

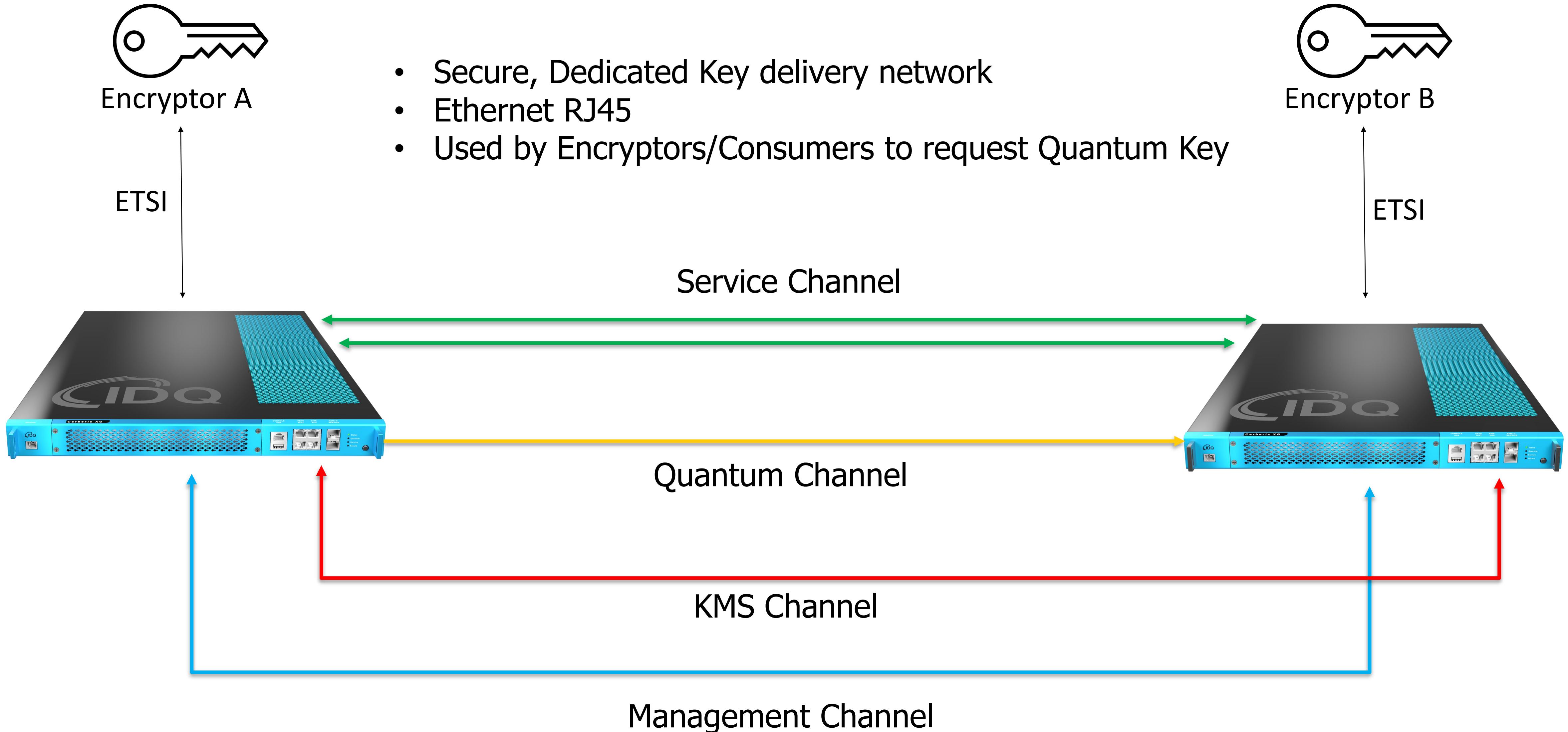


# QKD Topology

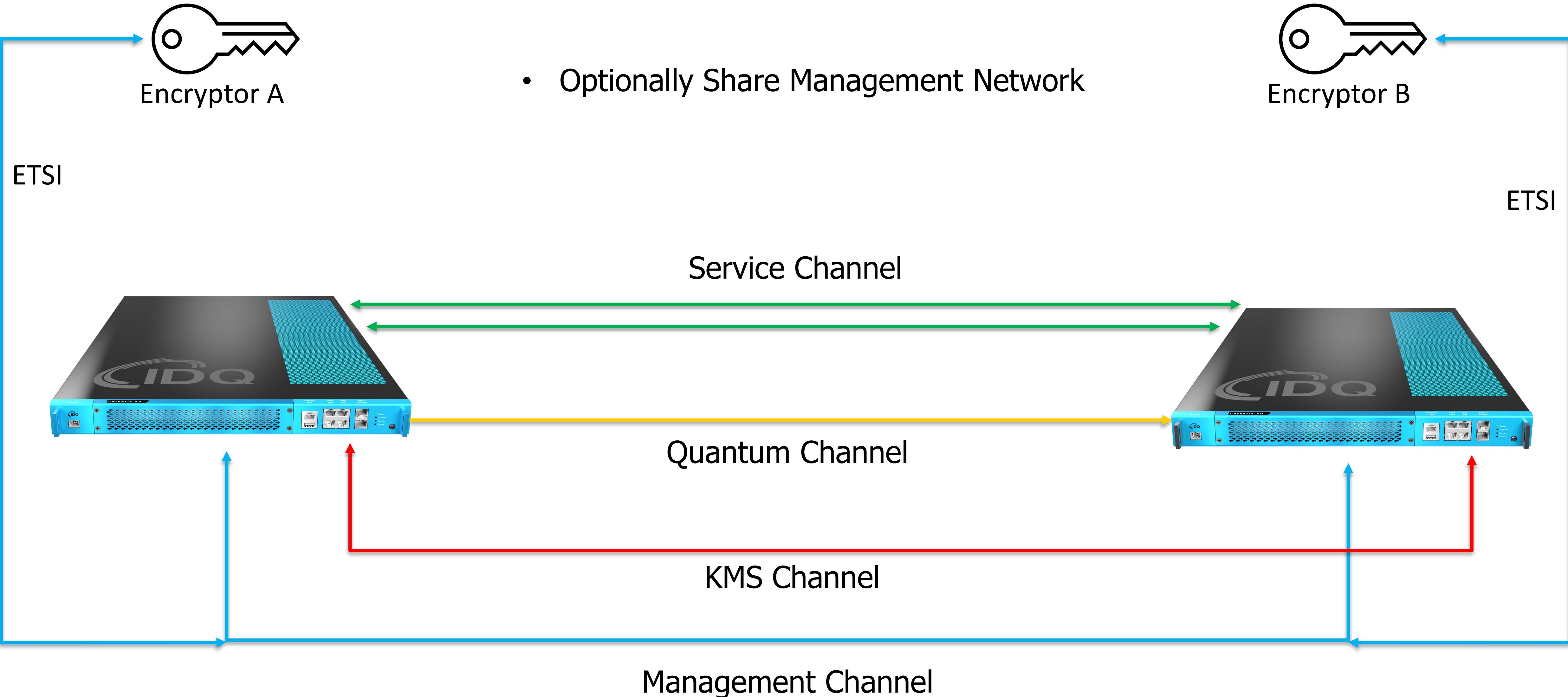
- Optionally separate KMS traffic to dedicated network for added security
- RJ45 or Optical SFP



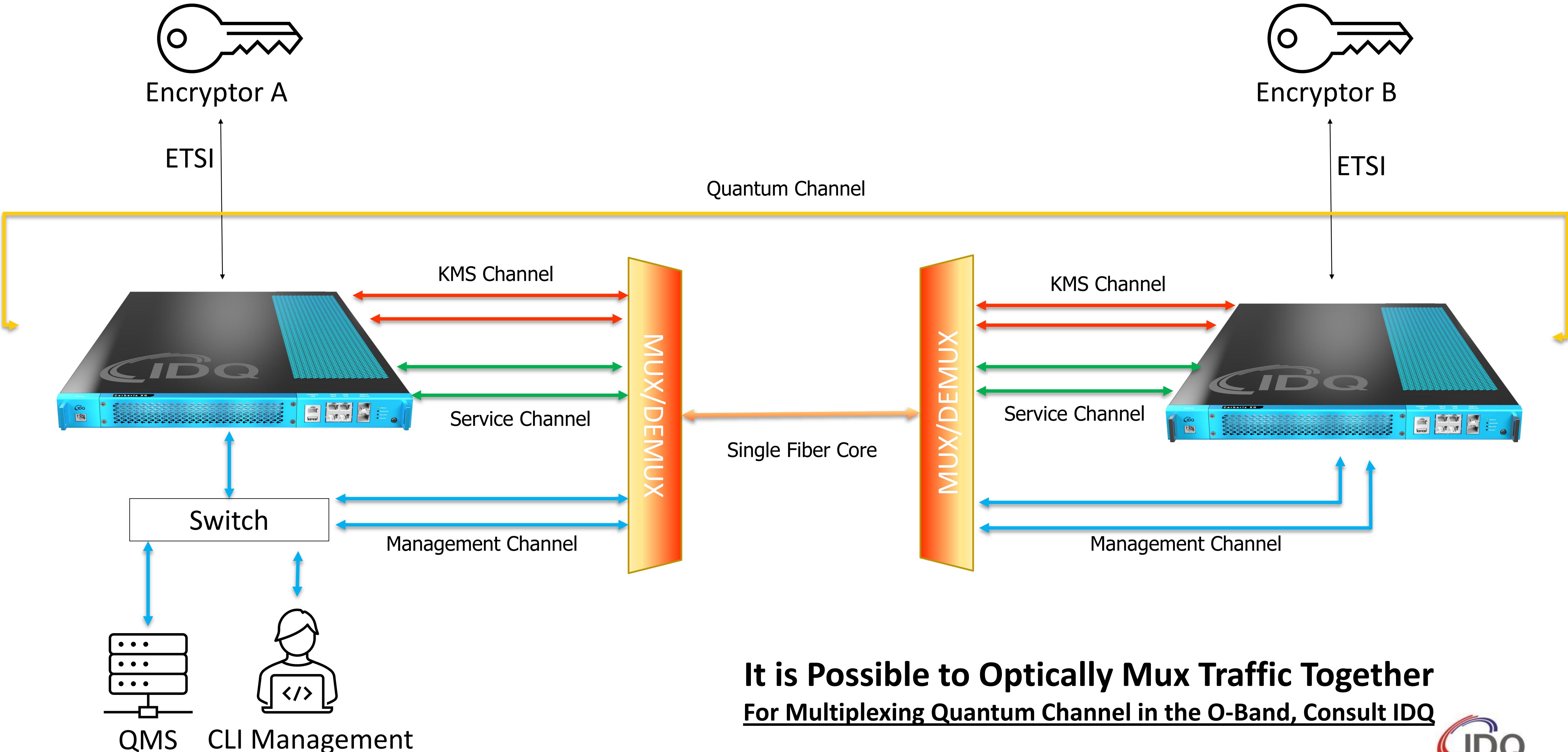
# QKD Topology



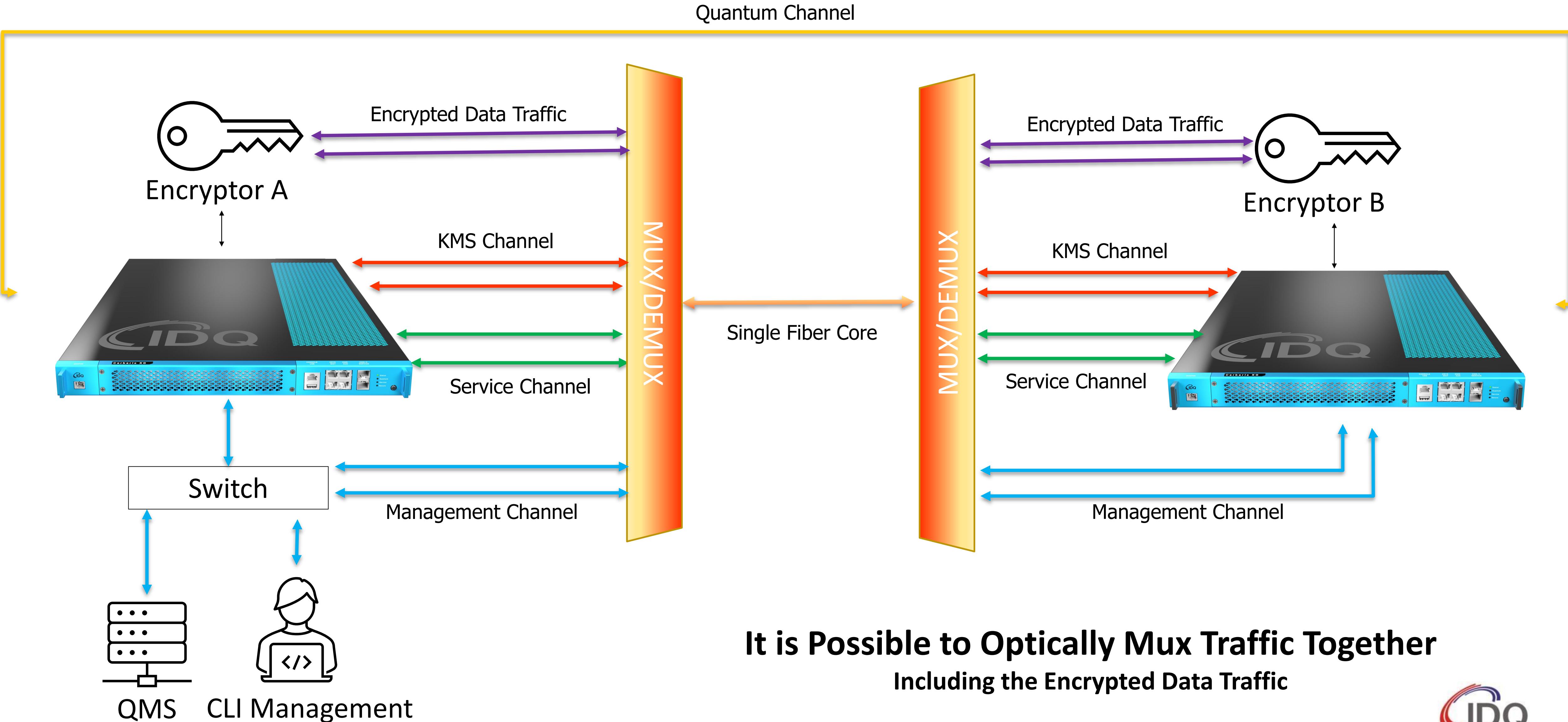
# QKD Topology



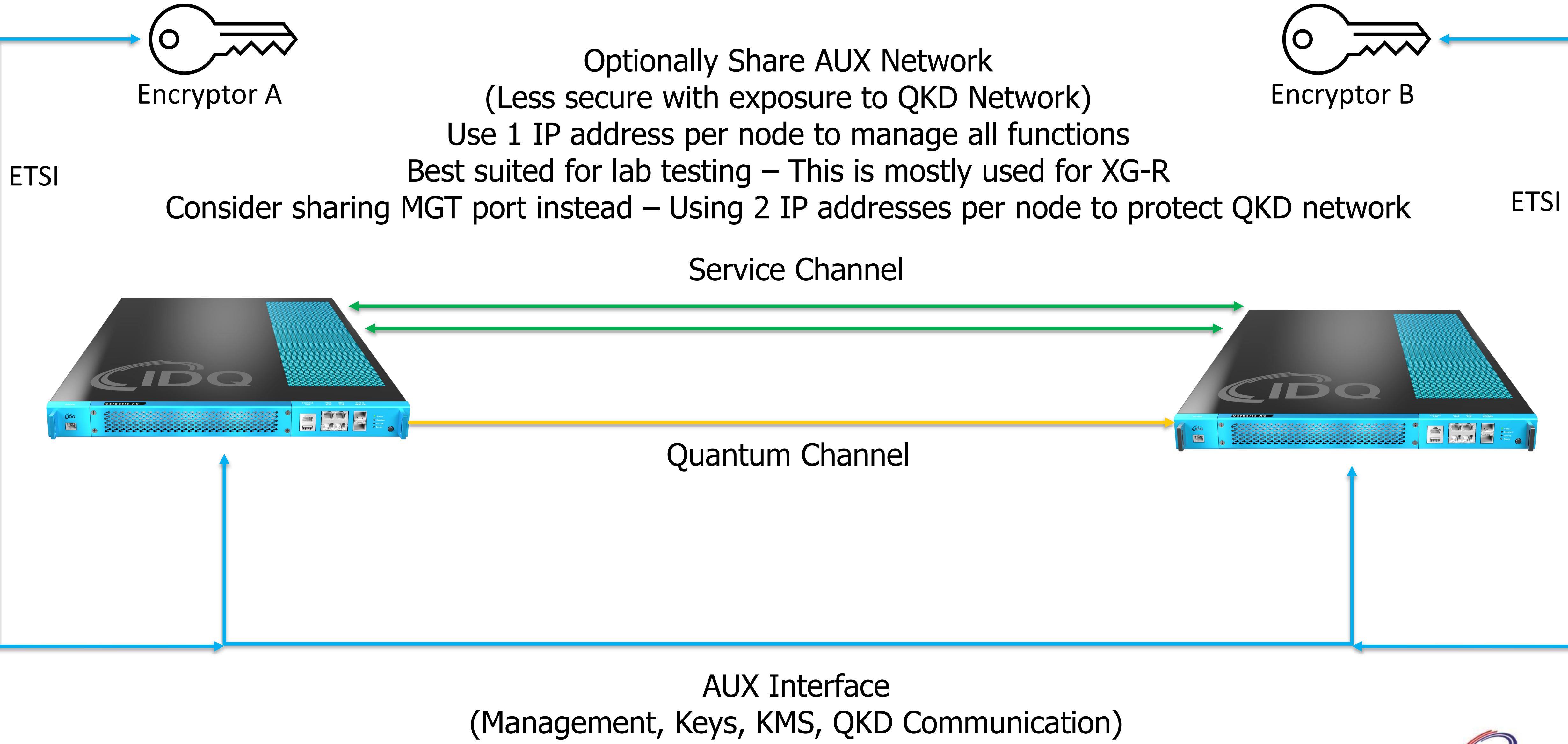
# QKD Topology



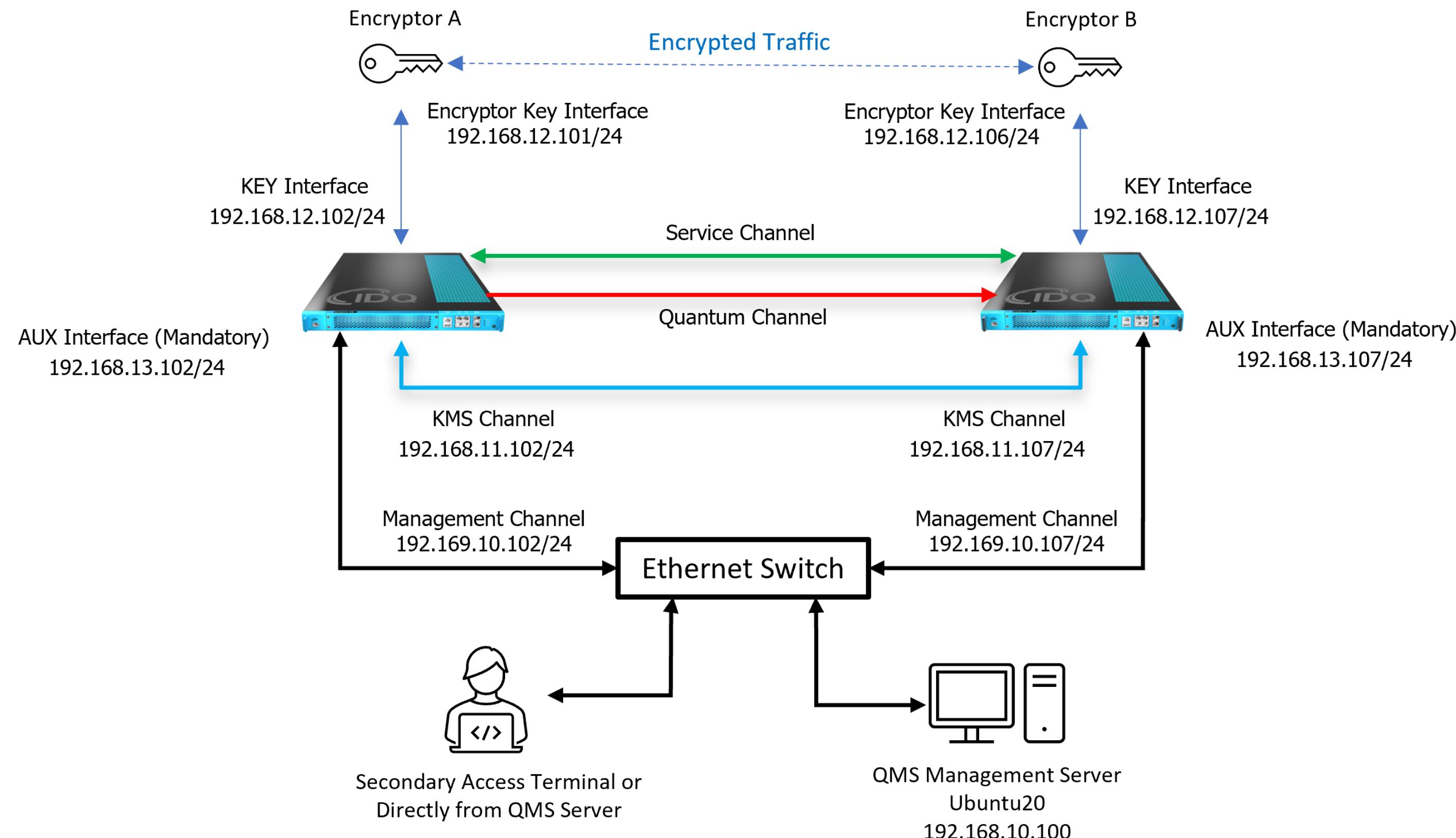
# QKD Topology



# QKD Topology

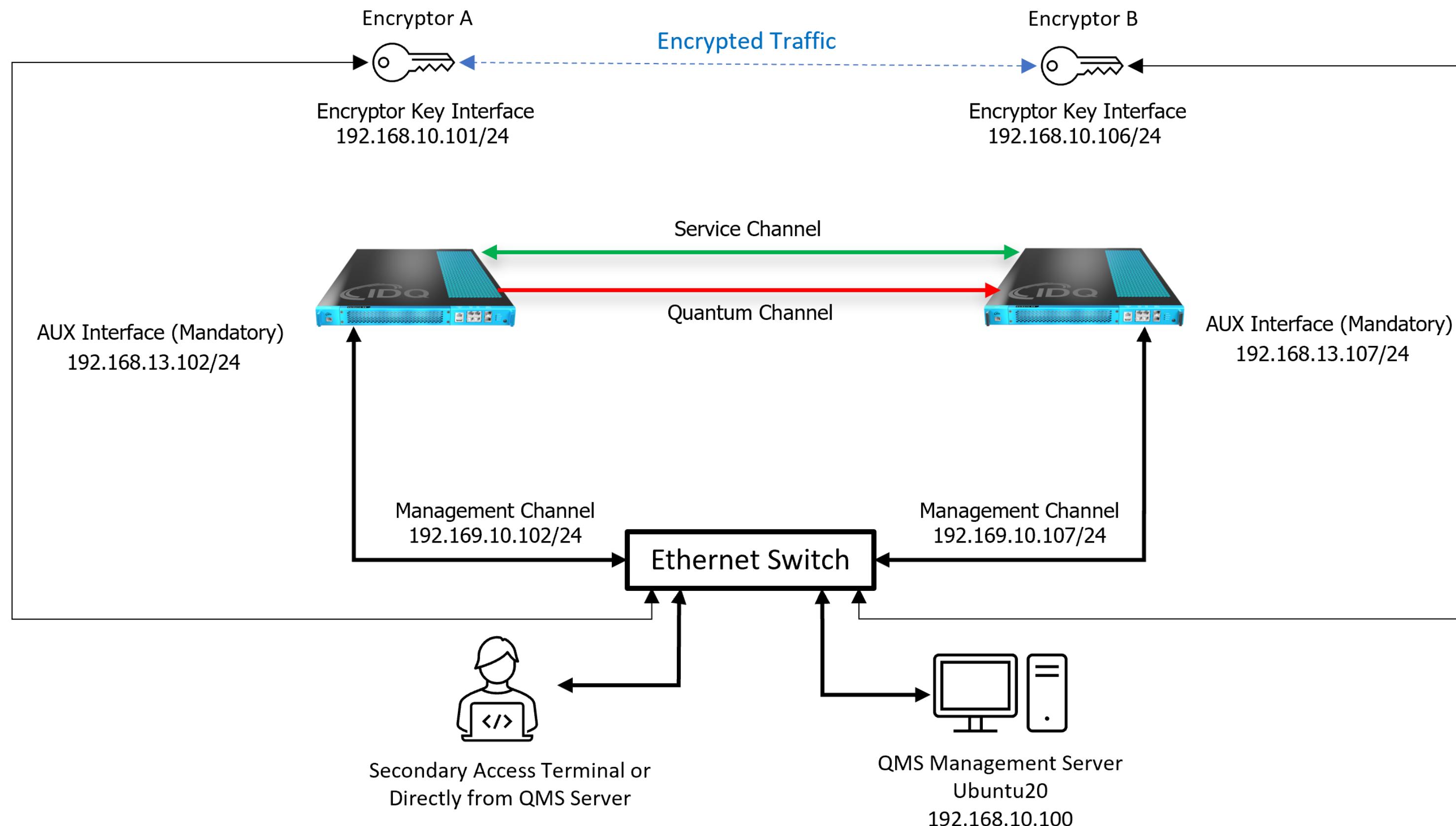


# Separate Networks for Management, Key, KMS, AUX

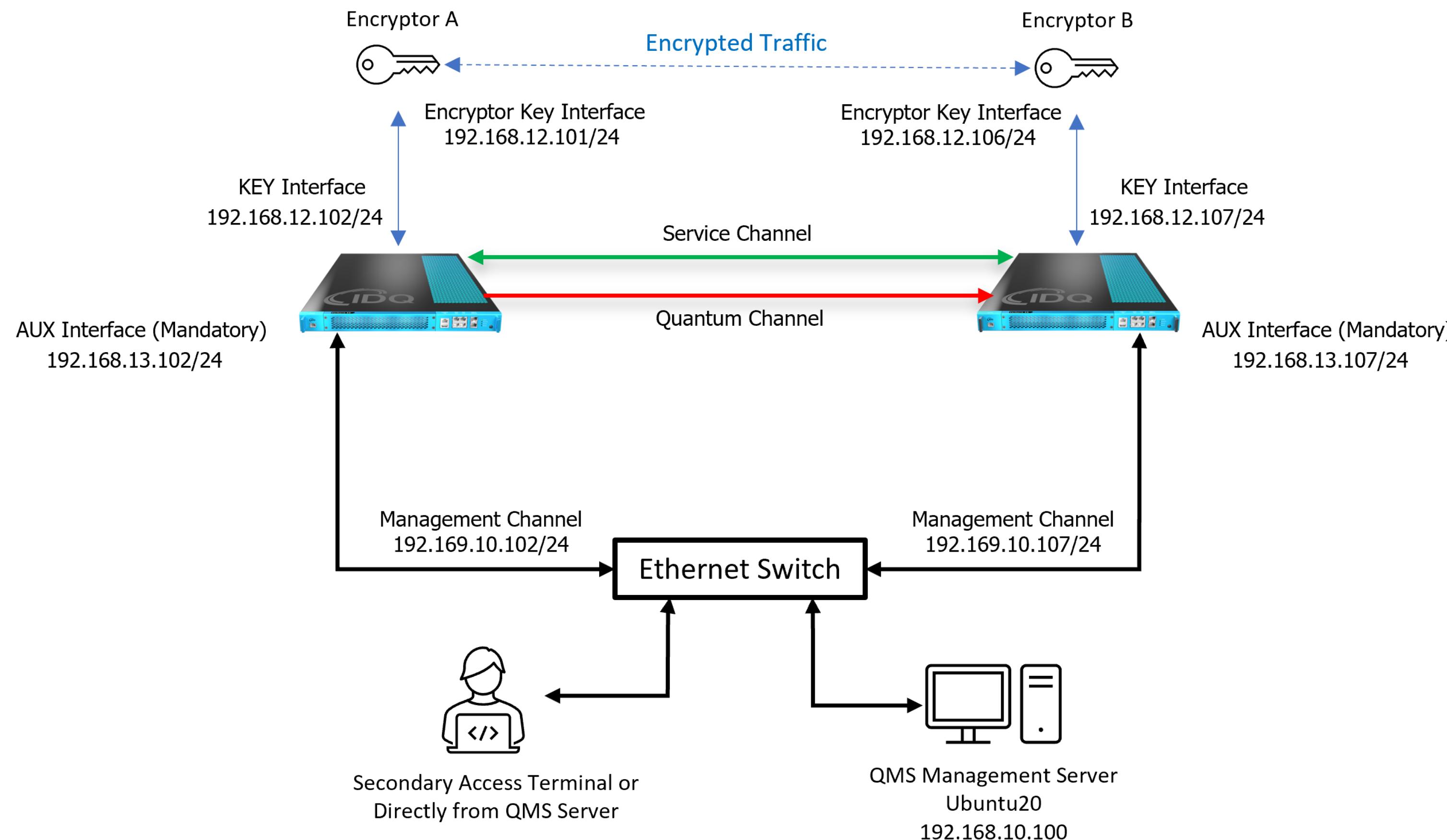


MGT Alice:	192.168.10.102/24
KMS Alice:	192.168.11.102/24
KEY Alice:	192.168.12.102/24
AUX Alice:	192.168.13.102/24
Encryptor Alice:	192.168.12.101/24
MGT Bob:	192.168.10.107/24
KMS Bob:	192.168.11.107/24
KEY Bob:	192.168.12.107/24
AUX Bob:	192.168.13.107/24
Encryptor Bob:	192.168.12.106/24
QMS Server: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

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



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



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

# Default Route Limitation

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

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

Static routes are not supported.

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



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

## QKD Chassis Details

### **Goal of this section:**

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



CerberisXG - Commercial



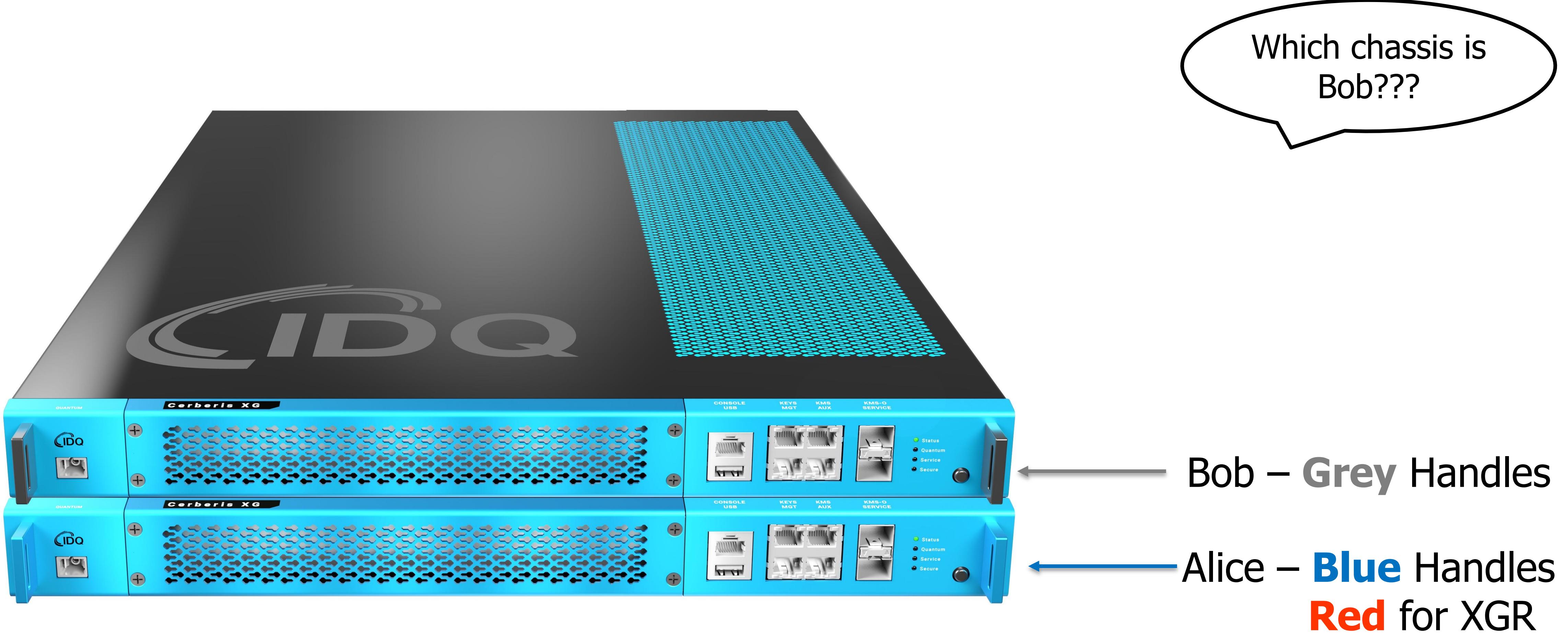
- **QSafe & Resilient KMS**
  - Interface with major encryptor vendors
  - Complex Network Topology support
- **QNet CLI (Embedded / Remote) interface**
  - Administration
  - Configuration
  - Monitoring
- **Web API for Configuration & Monitoring**
  - QKD
  - KMS
- **Support for Standard Monitoring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

CerberisXGR - Academic



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

# Bob vs Alice

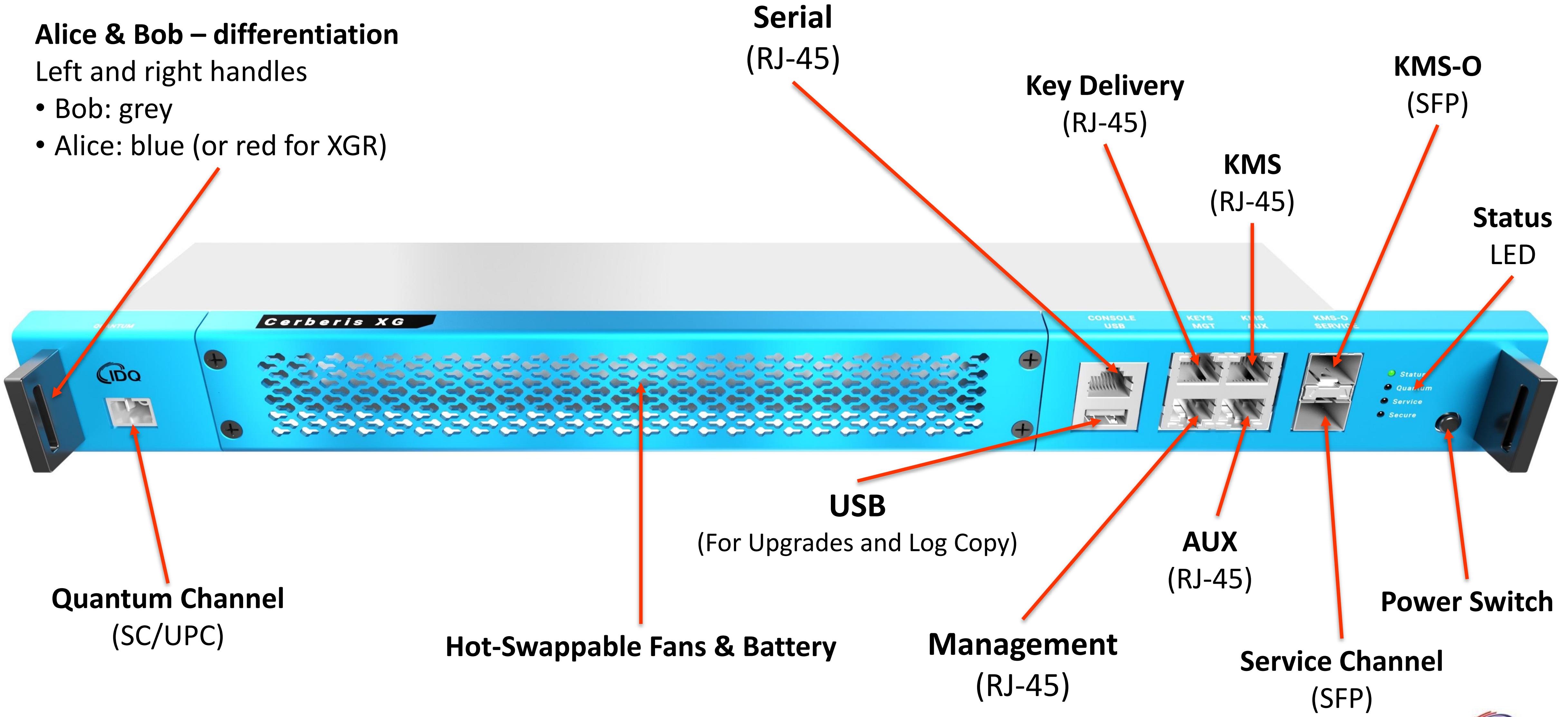


# Front Panel

## Alice & Bob – differentiation

Left and right handles

- Bob: grey
- Alice: blue (or red for XGR)

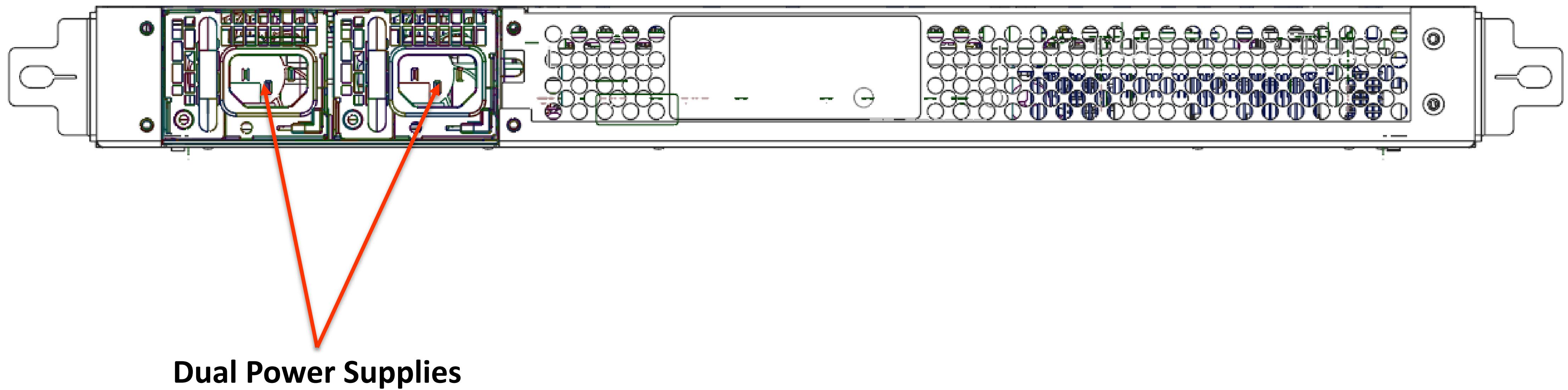


# Status LEDs



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

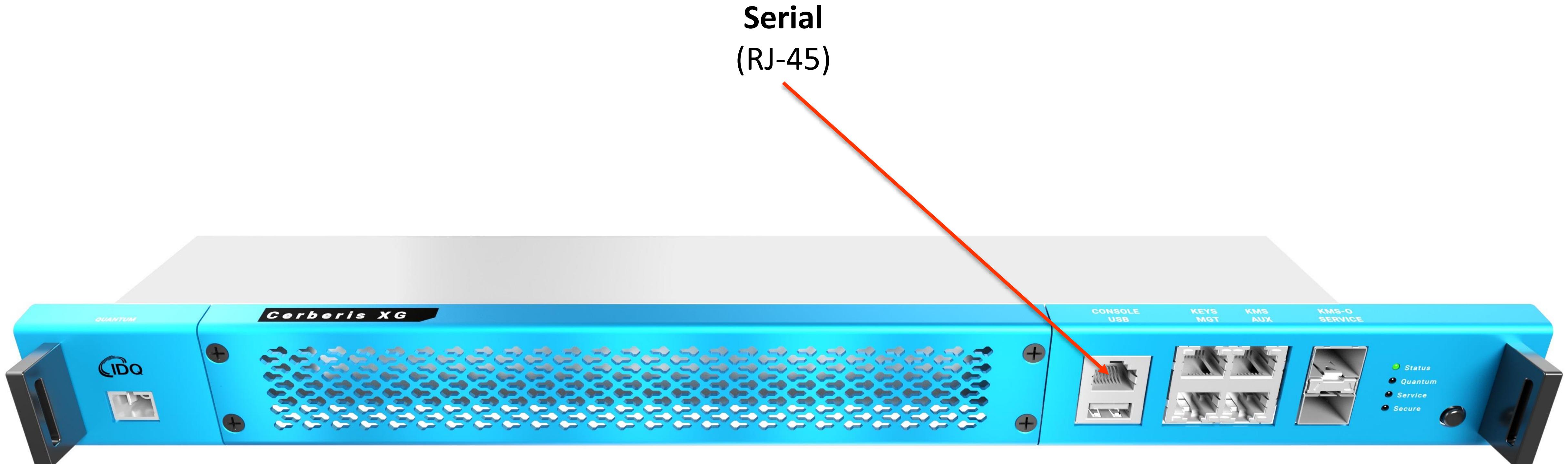
Rear



**Dual Power Supplies**

# Serial Interface

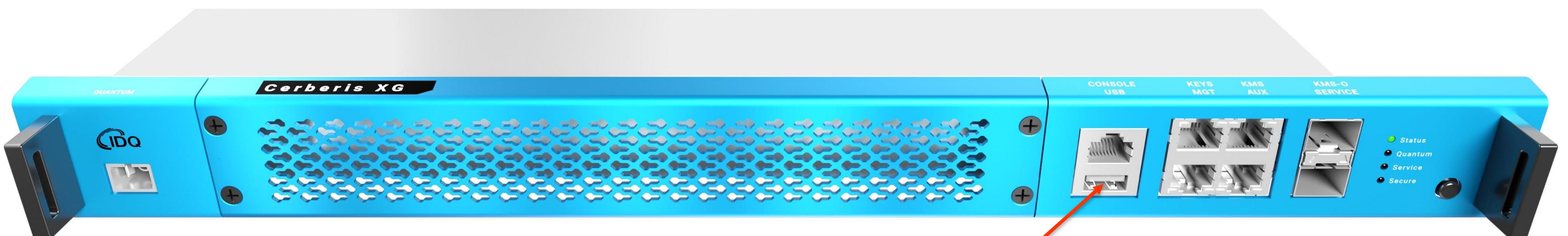
SWISS  
QUANTUM+



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

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

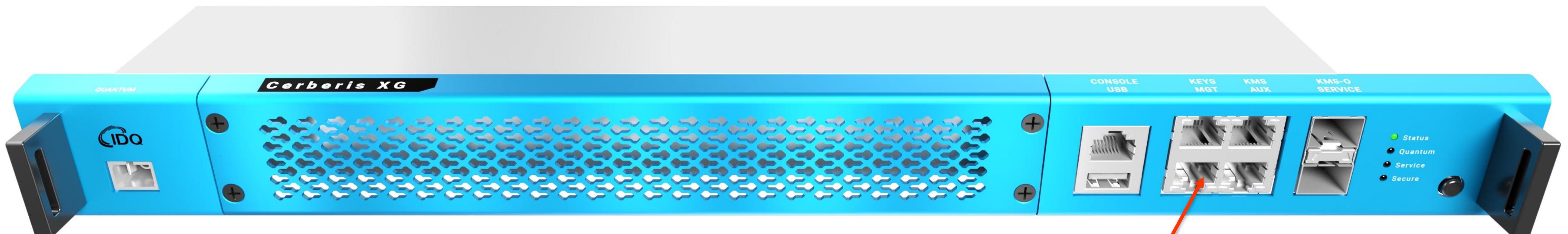
# Serial Interface



## USB Interface:

- System Upgrades
- Copy debug logs

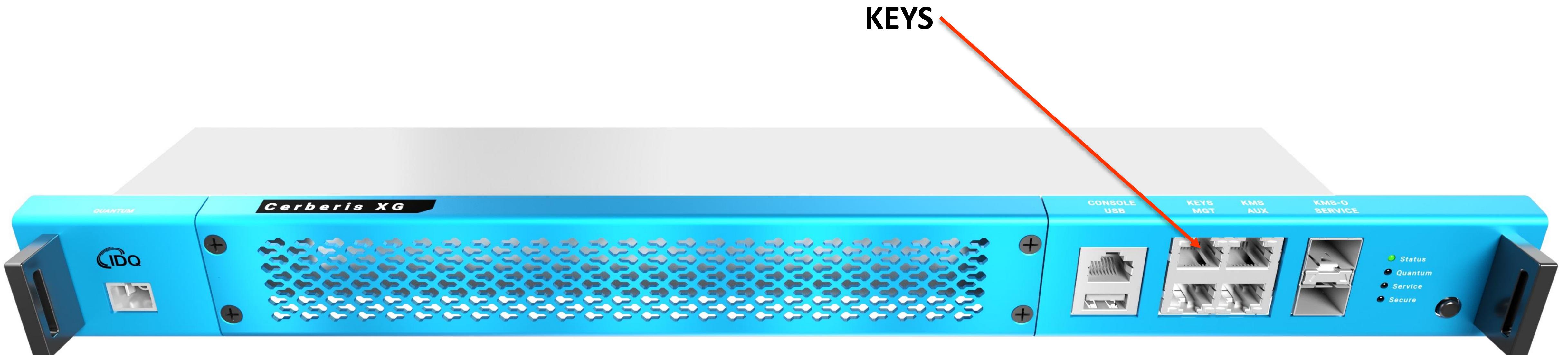
# MGT Interface



## Management Interface

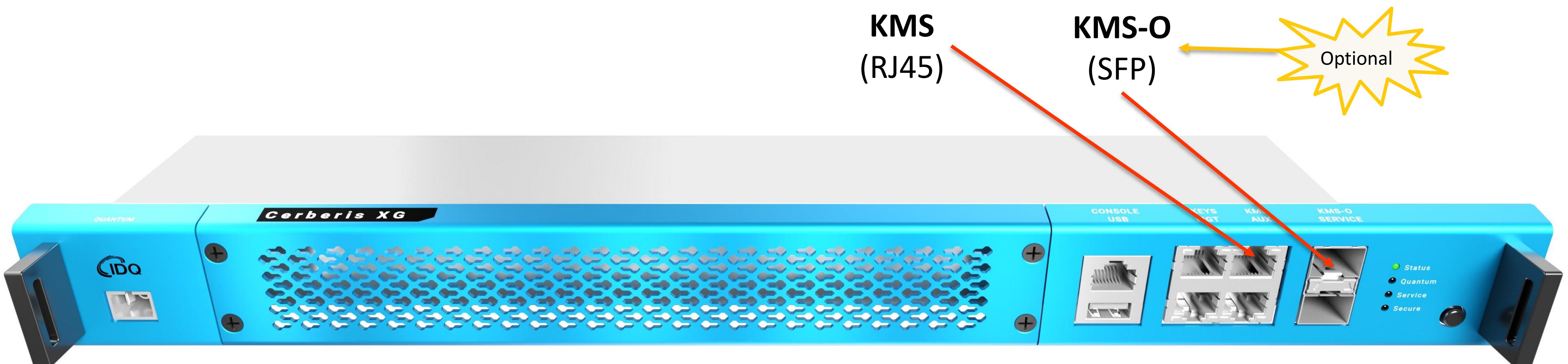
- SSH Access to CLI
- QMS/QNET Management Interface
- Optional KMS to KMS Communication
- SNMP Monitoring
- Syslog Forwarding
- Optional Key Delivery Interface

**MGT**  
(RJ-45)



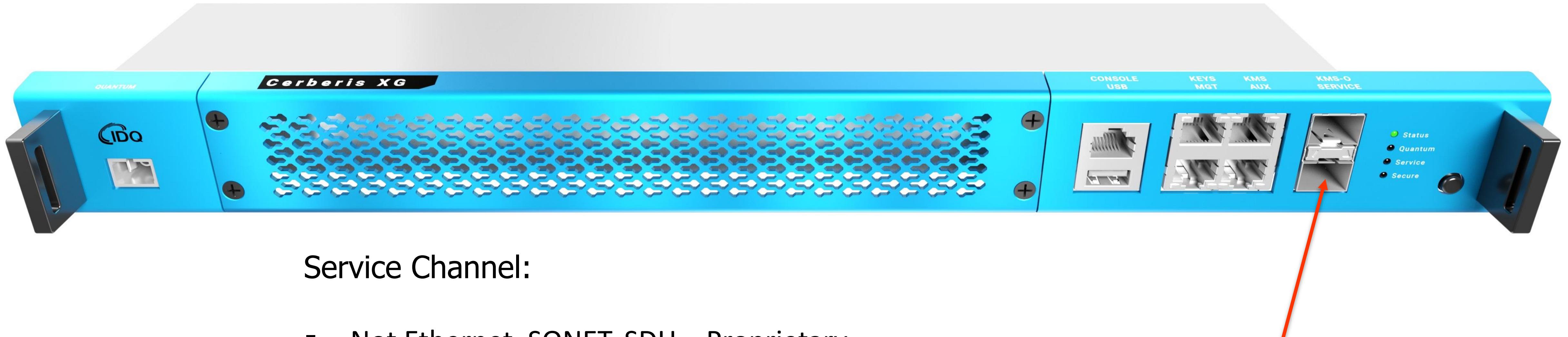
## Dedicated KEYS Interface:

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



## Dedicated KMS Interface (Key Management System):

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

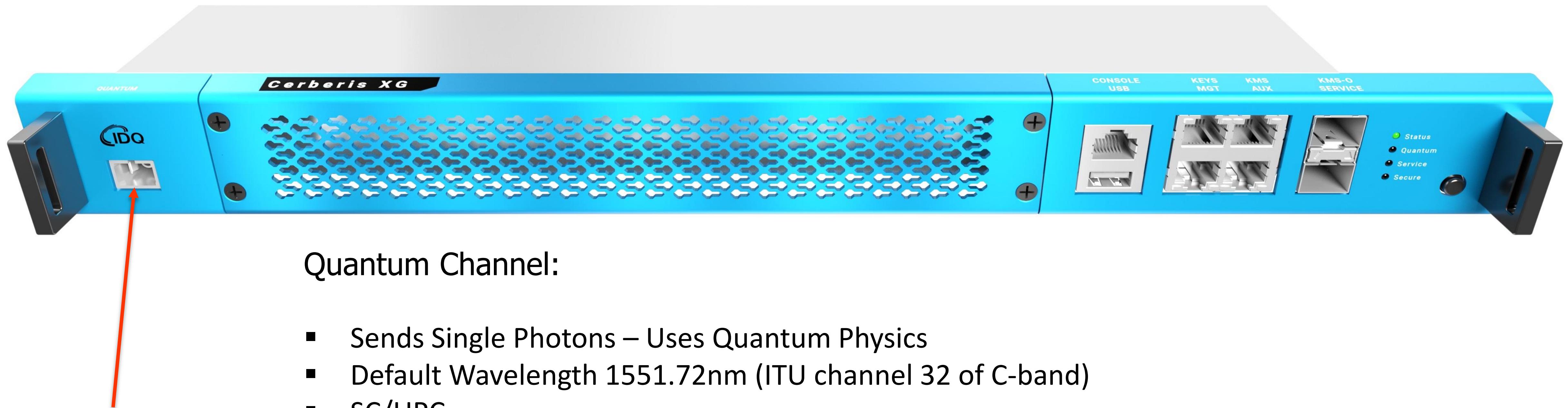


## Service Channel:

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

# Quantum Channel

SWISS  
QUANTUM+



**Quantum Channel  
(SC/UPC)**

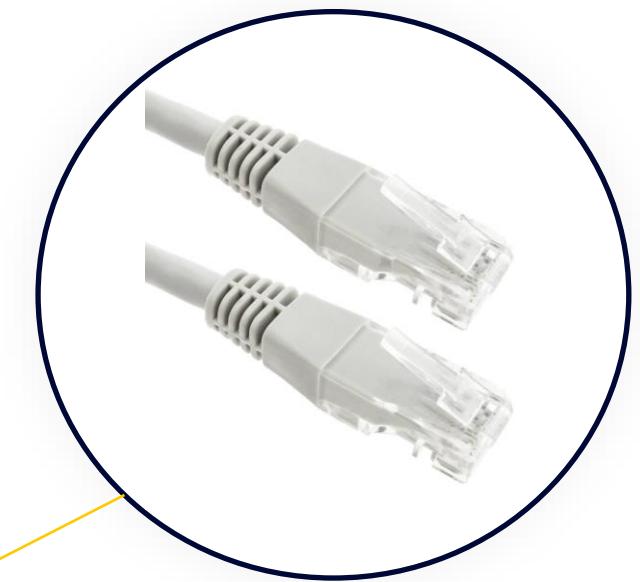
## Quantum Channel:

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

# AUX Interface – Trusted Node

**Bob**

**Alice**



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

## AUX Interface:

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

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

# CERBERIS XG

## Physical Installation step by step

## Goal of this part

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



## Safety precautions

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

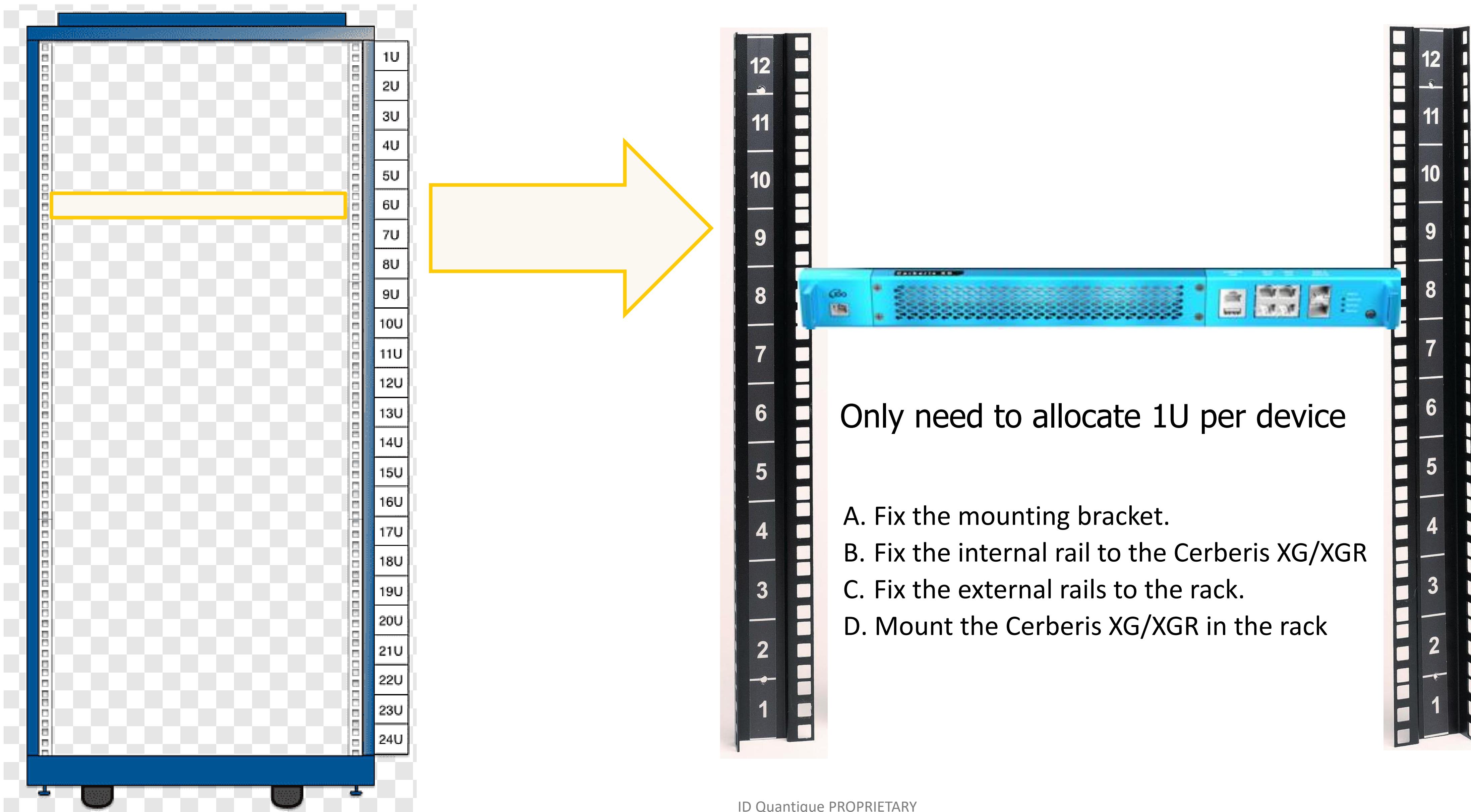
# Delivery

SWISS  
QUANTUM+

Verify the parcel is not damaged

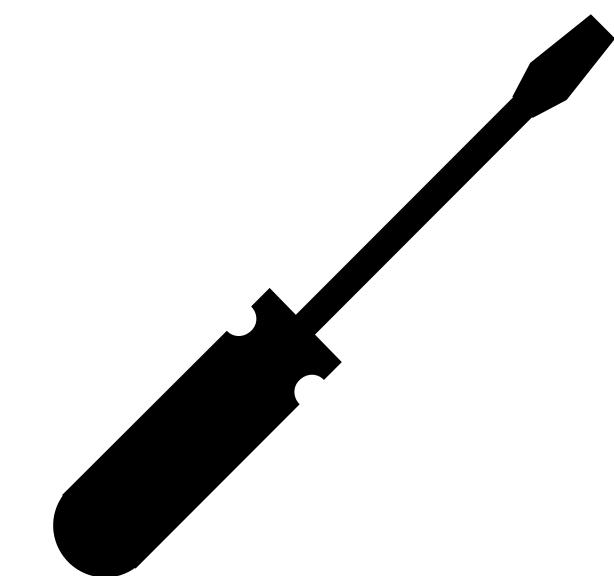
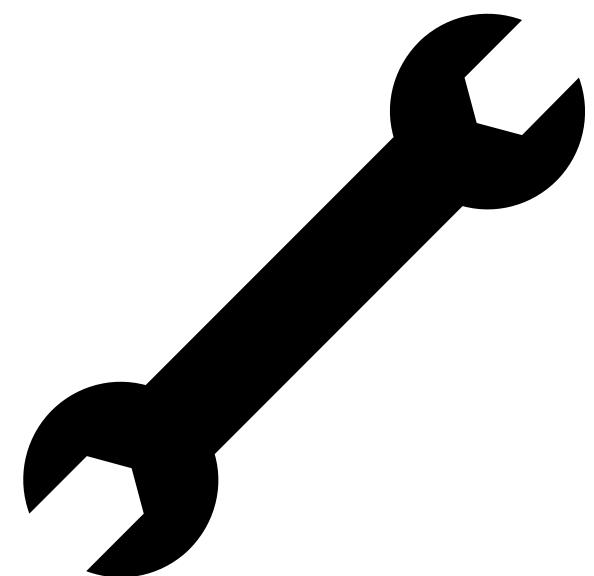


# Cerberis XG – Physical setup (Rack mount)



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

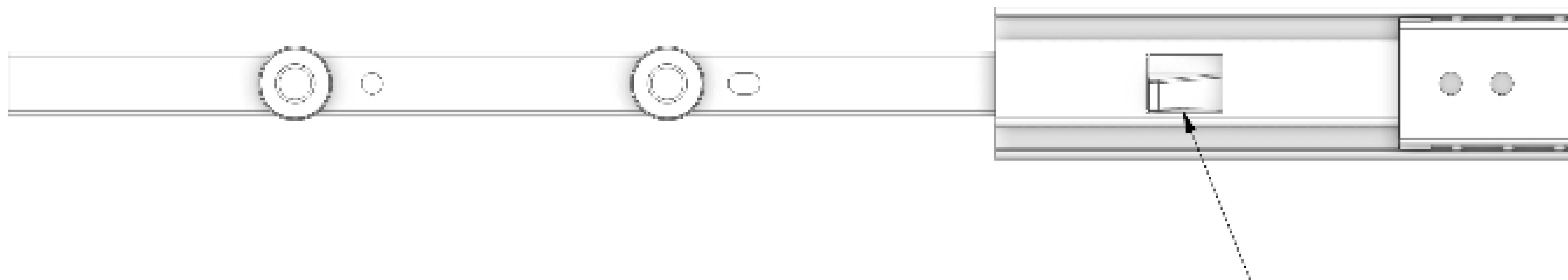
- Screwdriver Torx T10
- Screwdriver Torx T20
- Screwdriver Philips Head PH2
- 8mm wrench



# Cerberis XG – Physical setup (Rack mount)

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

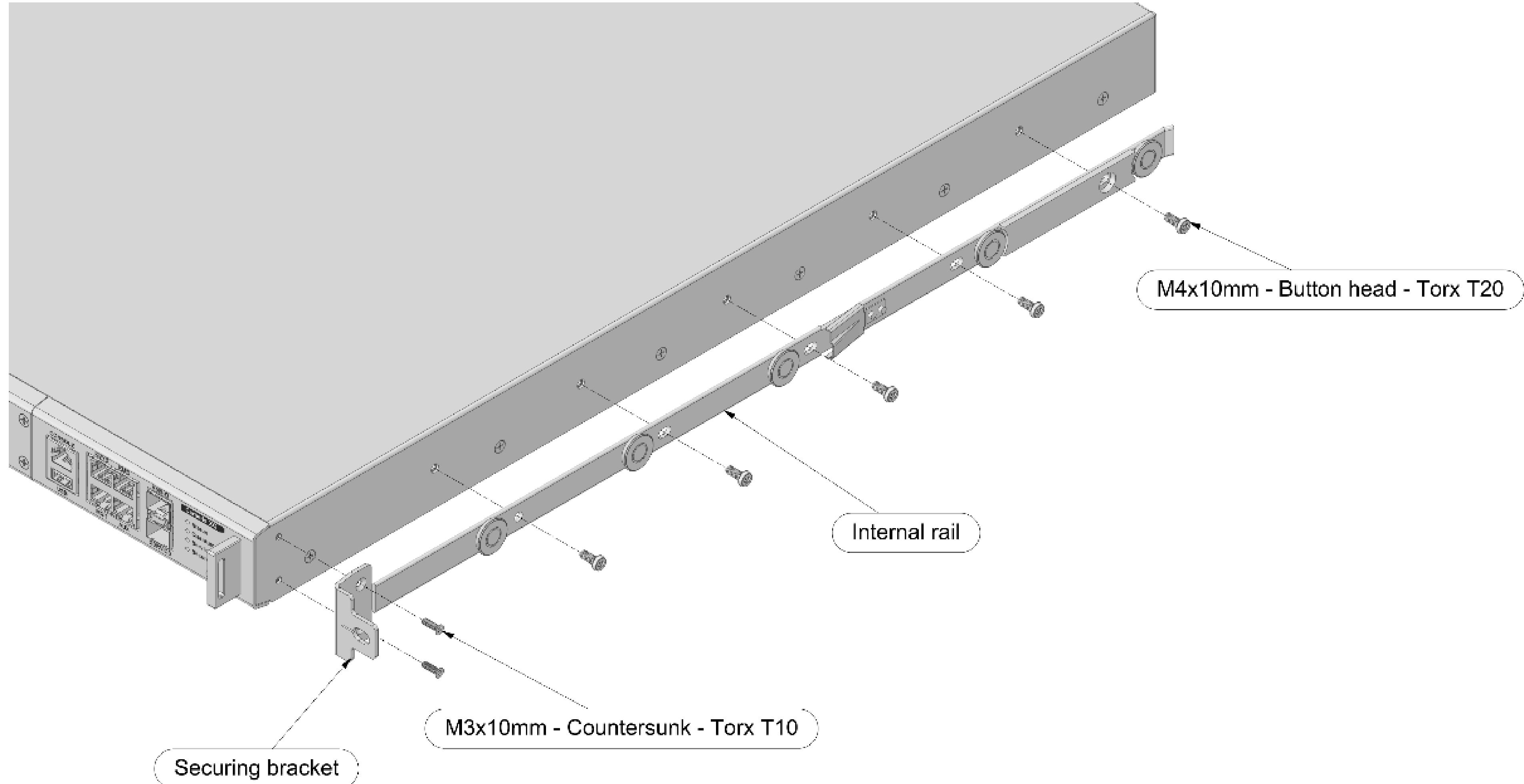
- 1) Disassemble the internal rails



Press on to extract internal rail

# Cerberis XG – Physical setup (Rack mount)

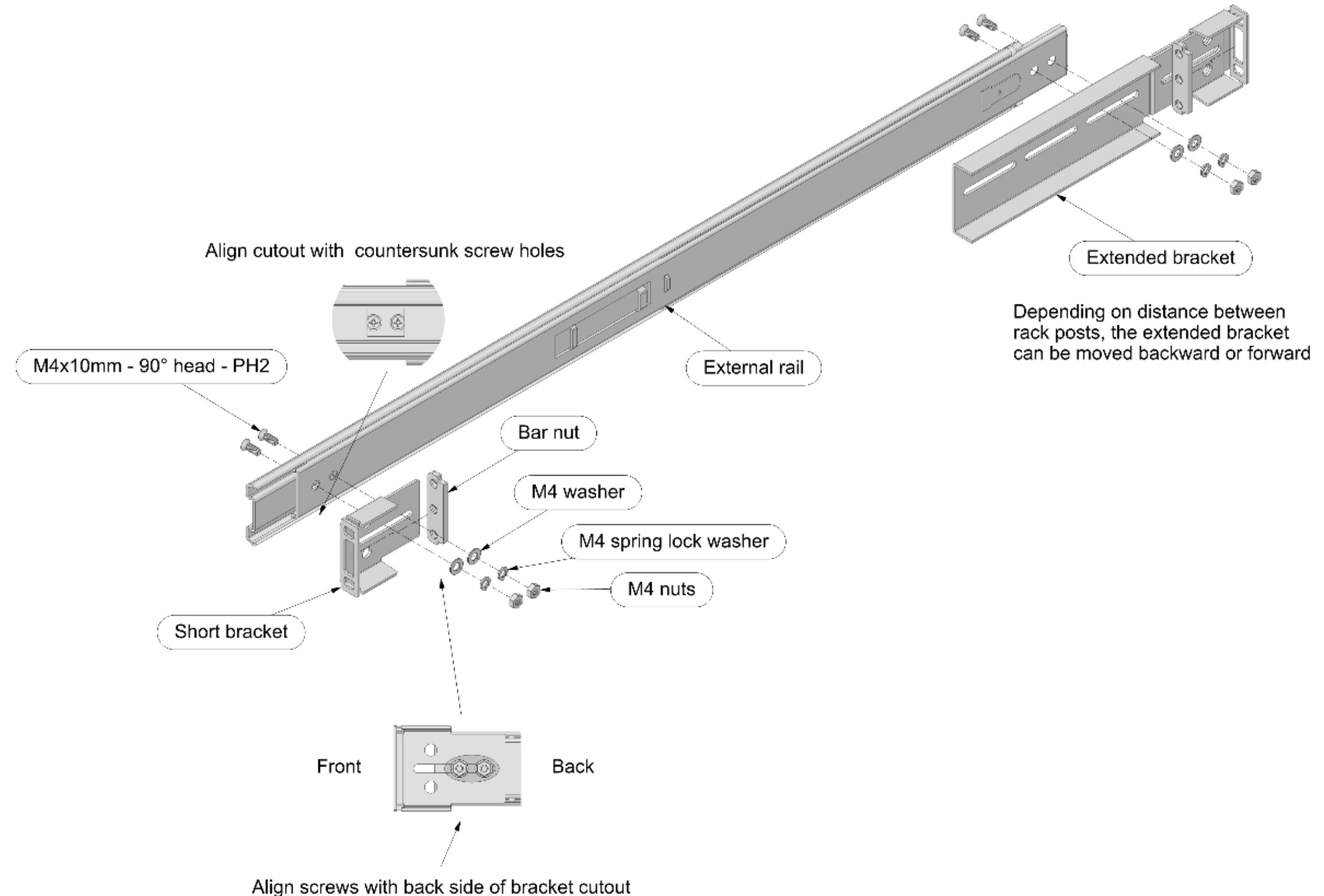
2) Fix the securing brackets and the internal rails to the CerberisXG



# Cerberis XG – Physical setup (Rack mount)

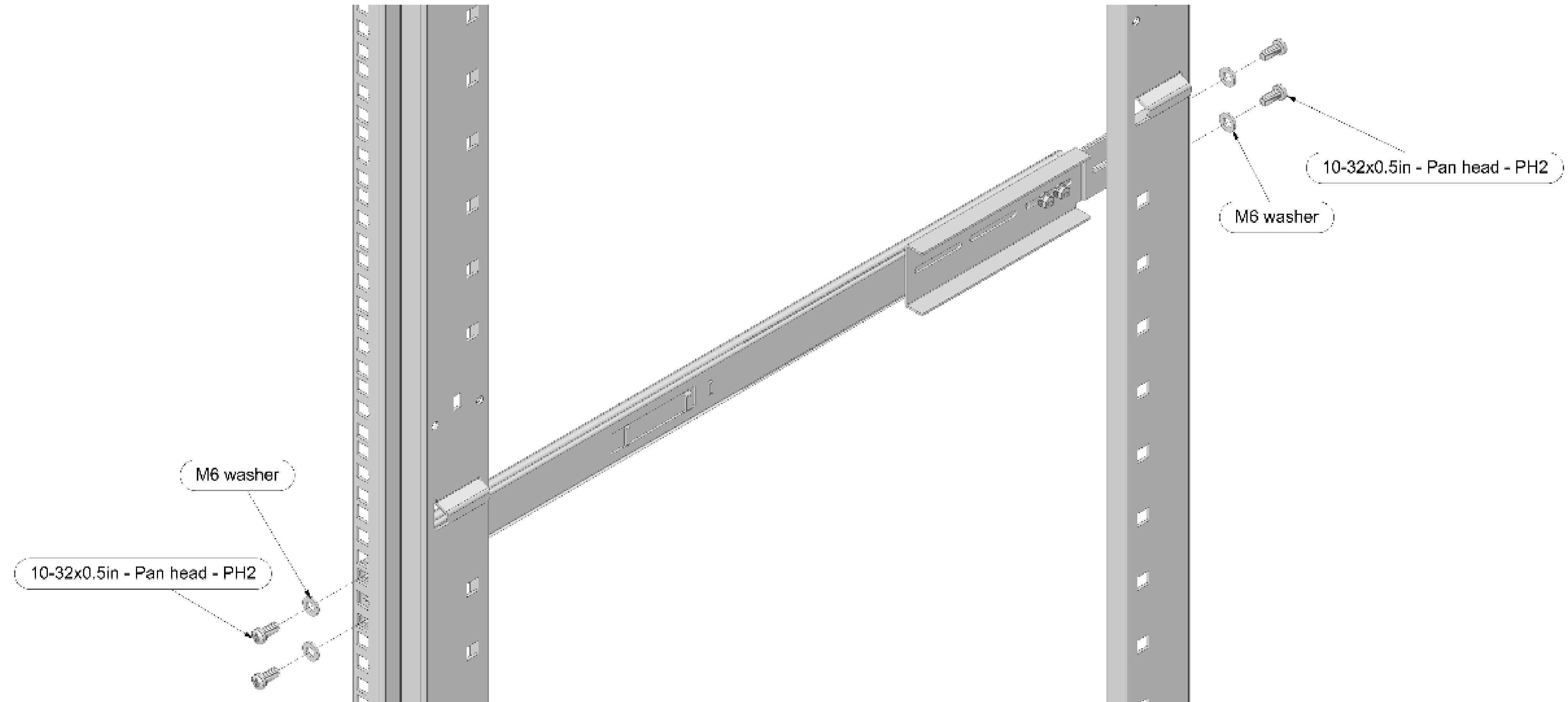
SWISS  
QUANTUM+

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



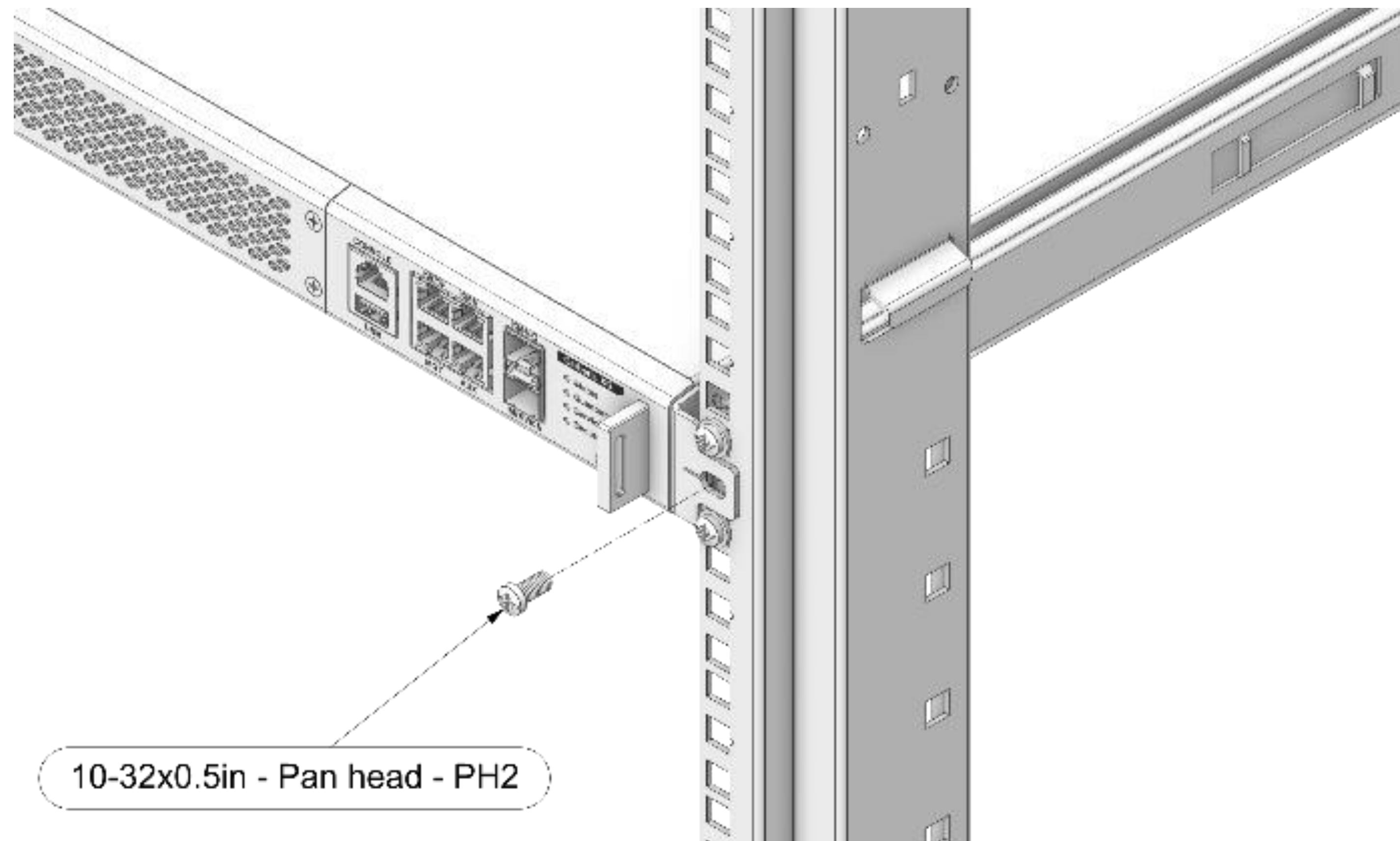
# Cerberis XG – Physical setup (Rack mount)

## 4) Fix the external rails in the rack



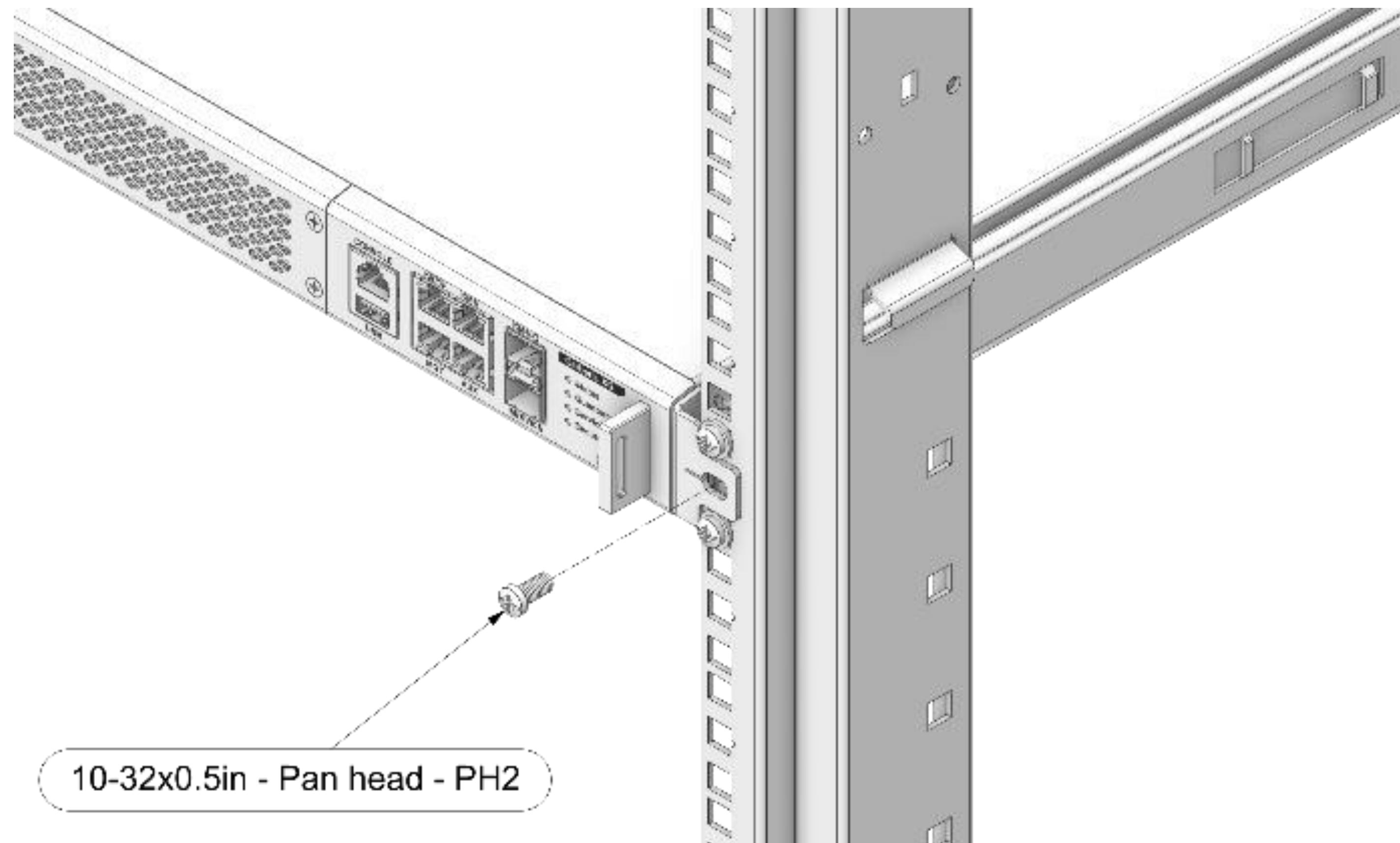
# Cerberis XG – Physical setup (Rack mount)

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



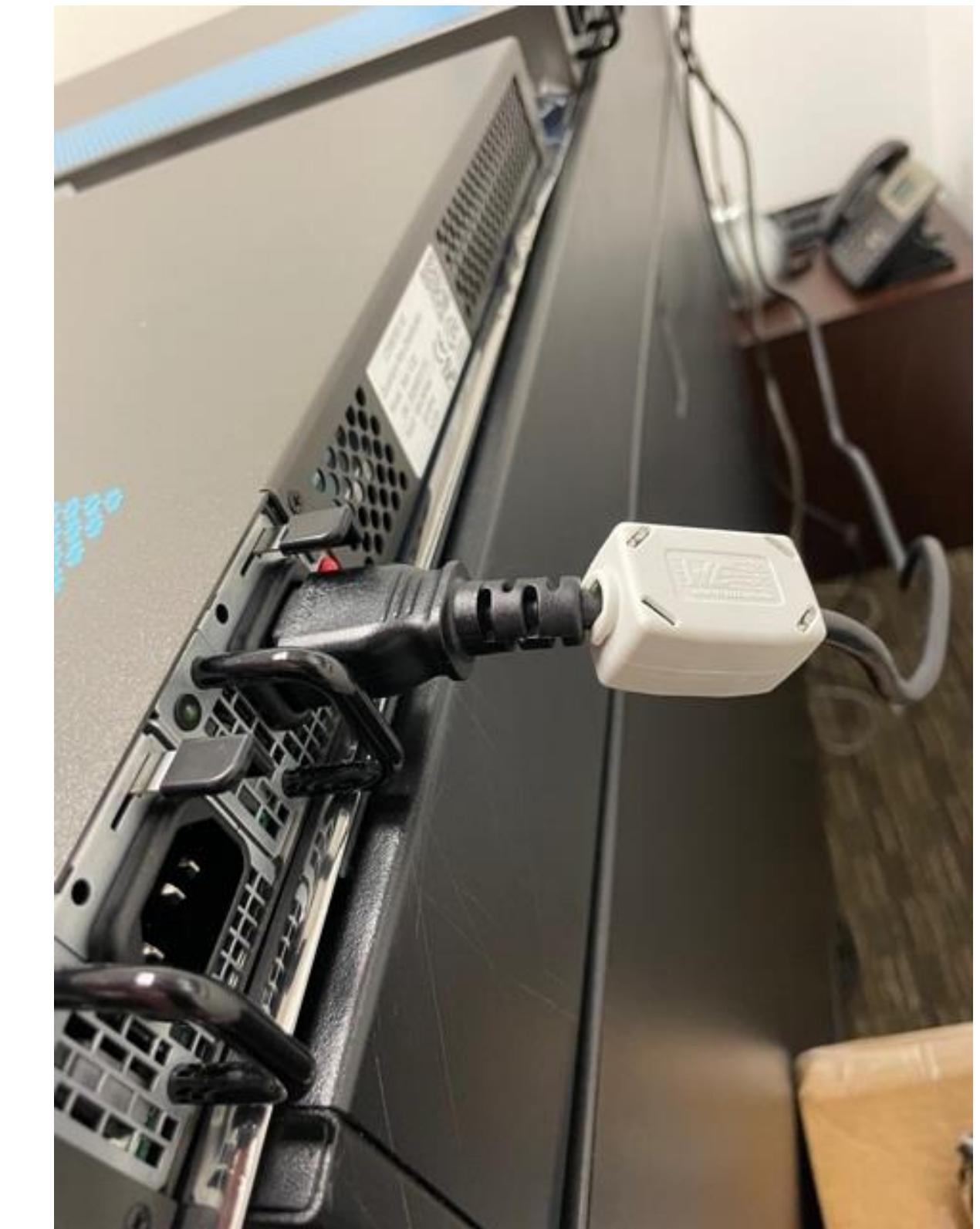
# Cerberis XG – Physical setup (Rack mount)

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



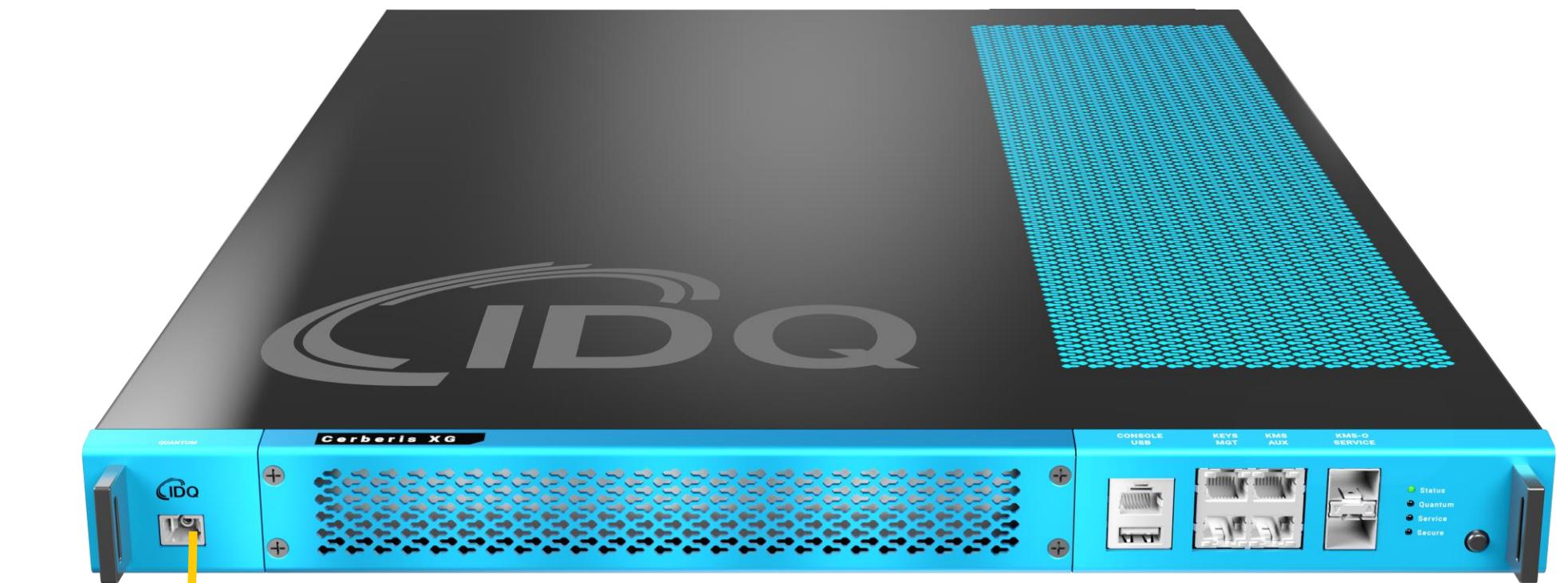
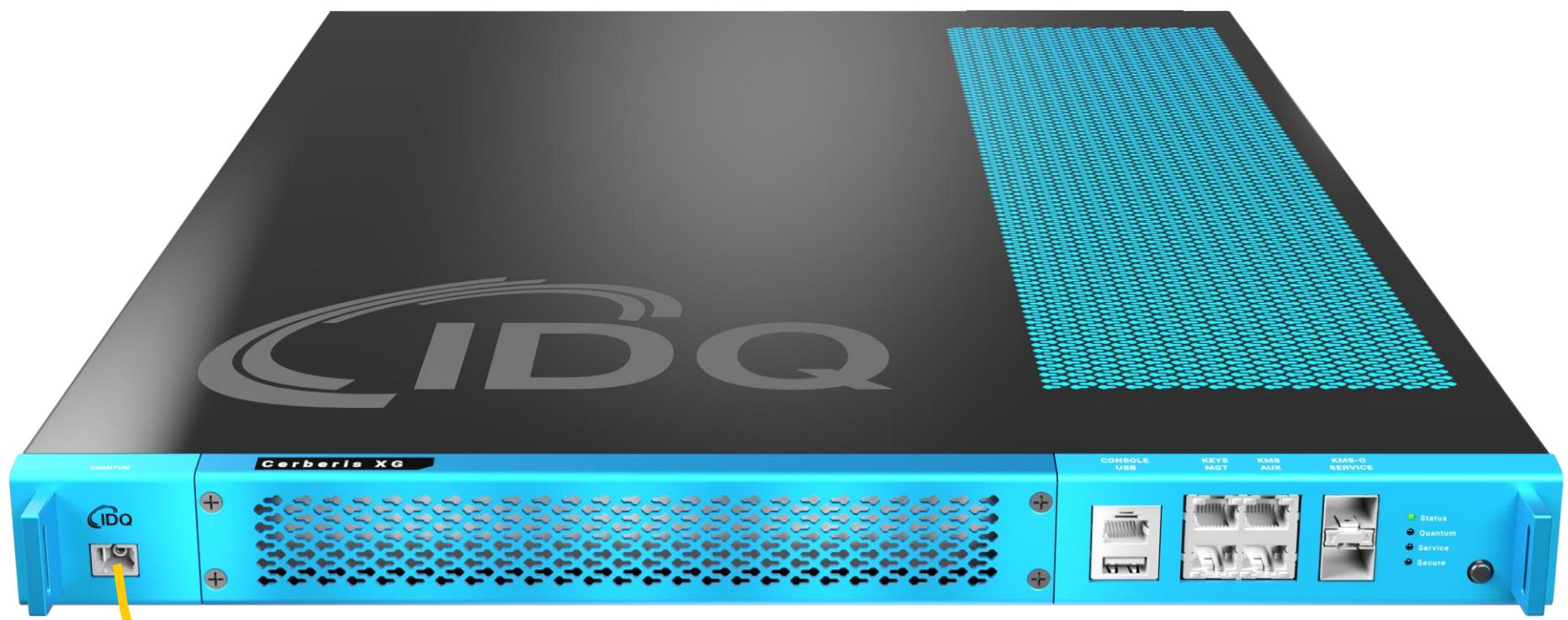
# Power Cables

Attach the ferrites to the power cables



# Cerberis XG – Physical setup (Quantum Channel)

## Connect Quantum Channel



SC/UPC SM fiber



If you are using a short fiber add fixed attenuator

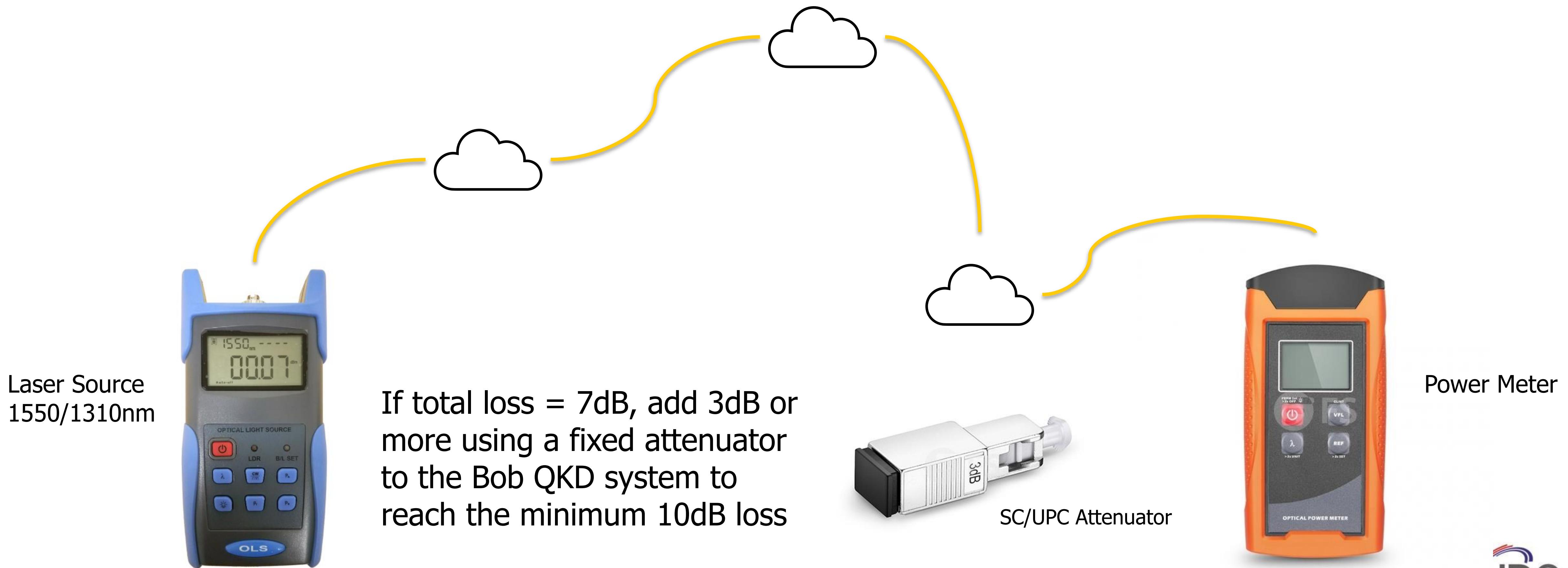


SC/UPC Attenuator (optional)

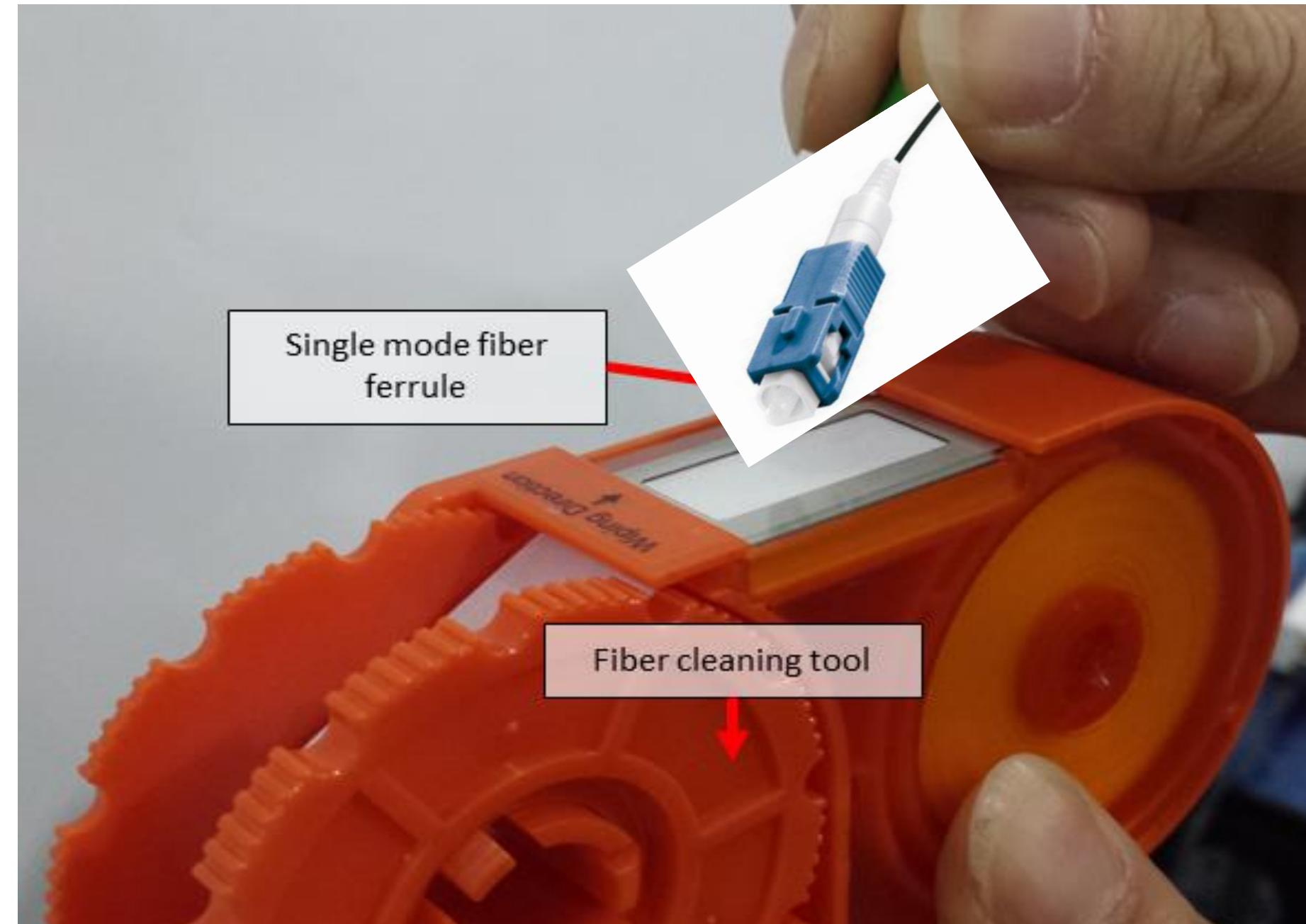
- ▶ The Quantum Channel requires a minimum loss of 10dB
- ▶ Maximum loss equals the rated budget for your system purchase
- ▶ For Example:
  - ⑩ A standard 12dB system must have losses between 10dB and 12dB
  - ⑩ A premium 18dB system must have losses between 10dB and 18dB
- ▶ Add the appropriate amount of attenuation as needed for your fiber connection.  
In a production environment, always measure the link loss using a power meter and laser source... or alternatively use an OTDR.
- ▶ If your fiber route has a natural loss of 7dB, add an additional 3dB attenuator to reach the minimum 10dB.

# Properly measure Quantum Channel Losses

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



# Cerberis XG – Physical setup (Quantum Channel)



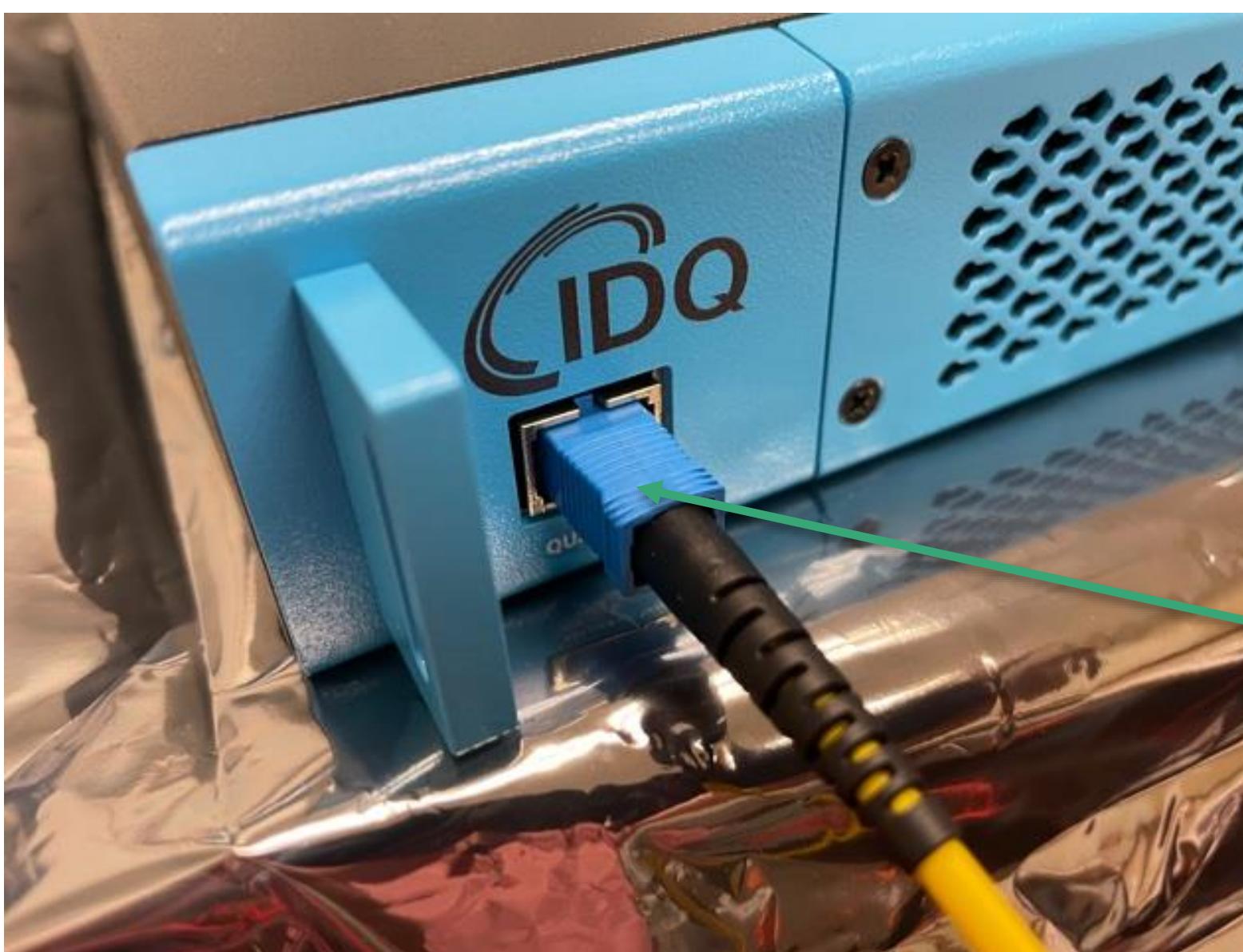
- Remove the protective cap from the SC/UPC connector end of the single mode fiber patch cord
- Clean the exposed fiber ferrule end by wiping it twice on the cleaning tool.
- Only wipe in the direction depicted on the cleaning tool!
- Always use a new/clean section of the cleaning tape

# Cerberis XG – Physical setup (Quantum Channel)



Attenuator

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



No Attenuator

# Physical Setup – Quantum Channel

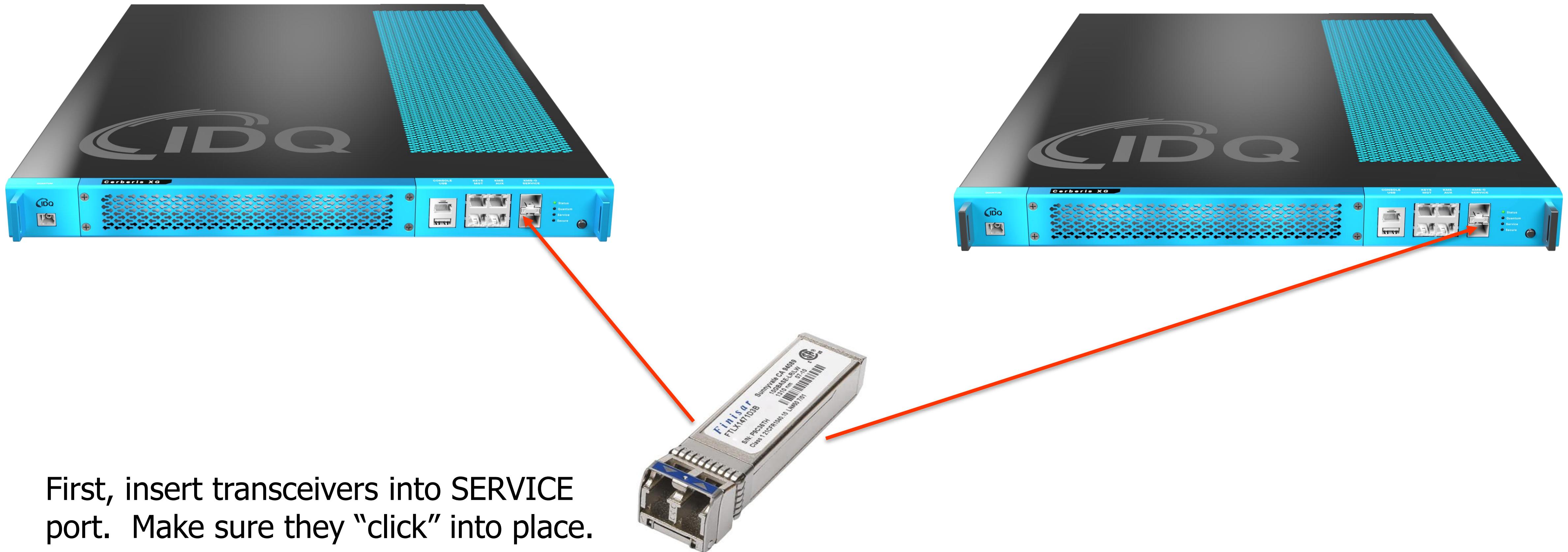
Optical connections, Quantum channel :

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

# Cerberis XG – Physical setup (Service Channel)

SWISS  
QUANTUM+

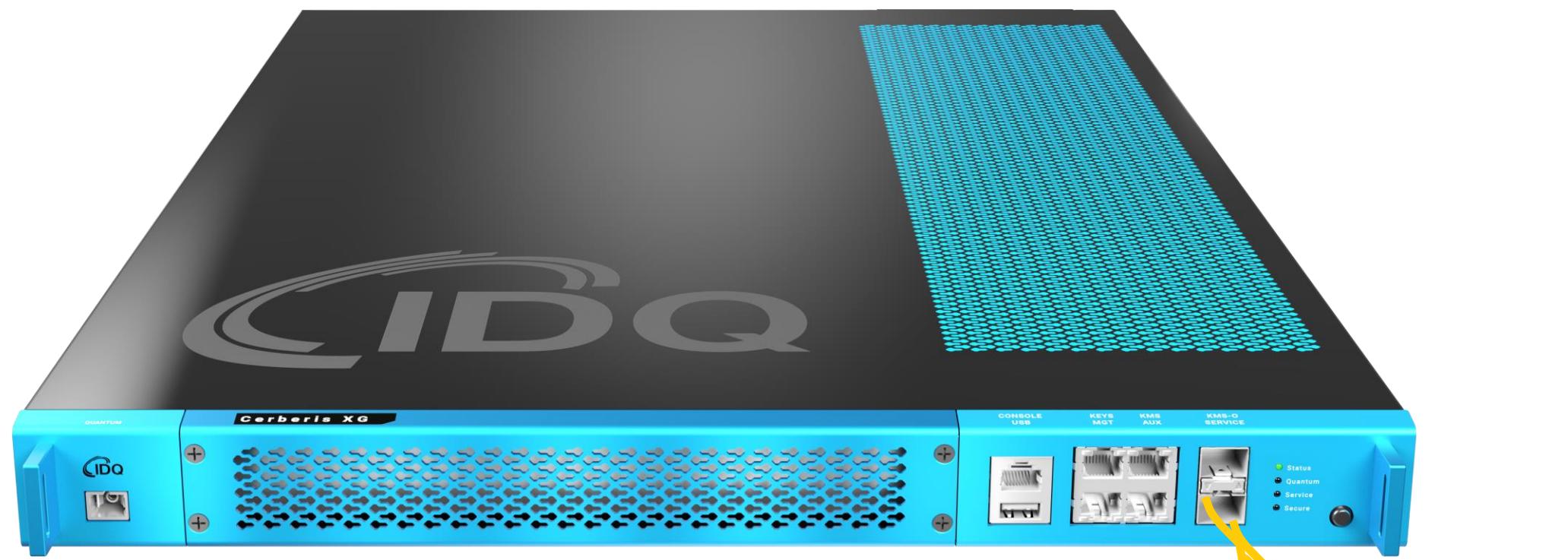
## Connect Service Channel



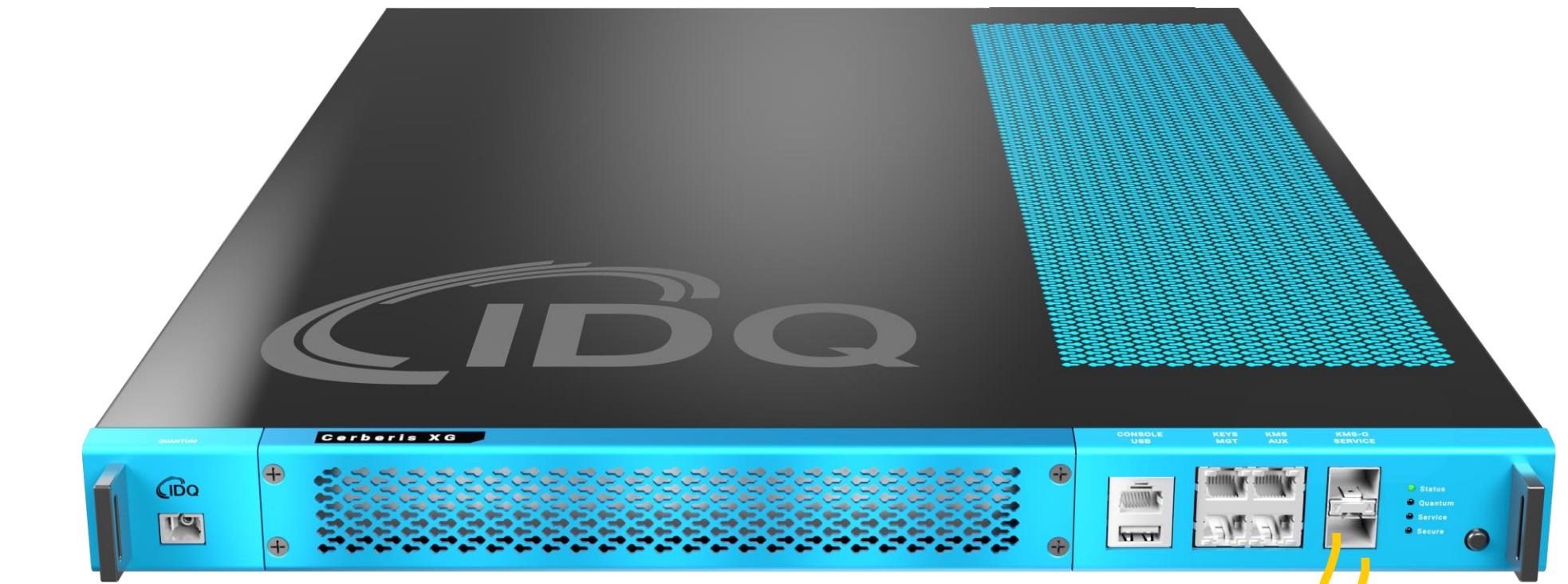
# Cerberis XG – Physical setup (Service Channel)

SWISS  
QUANTUM+

## Connect Service Channel



1) Clean fiber and attenuator tips



2) Connect duplex fibers to transceivers in SERVICE port

LC/UPC SMF fiber



+

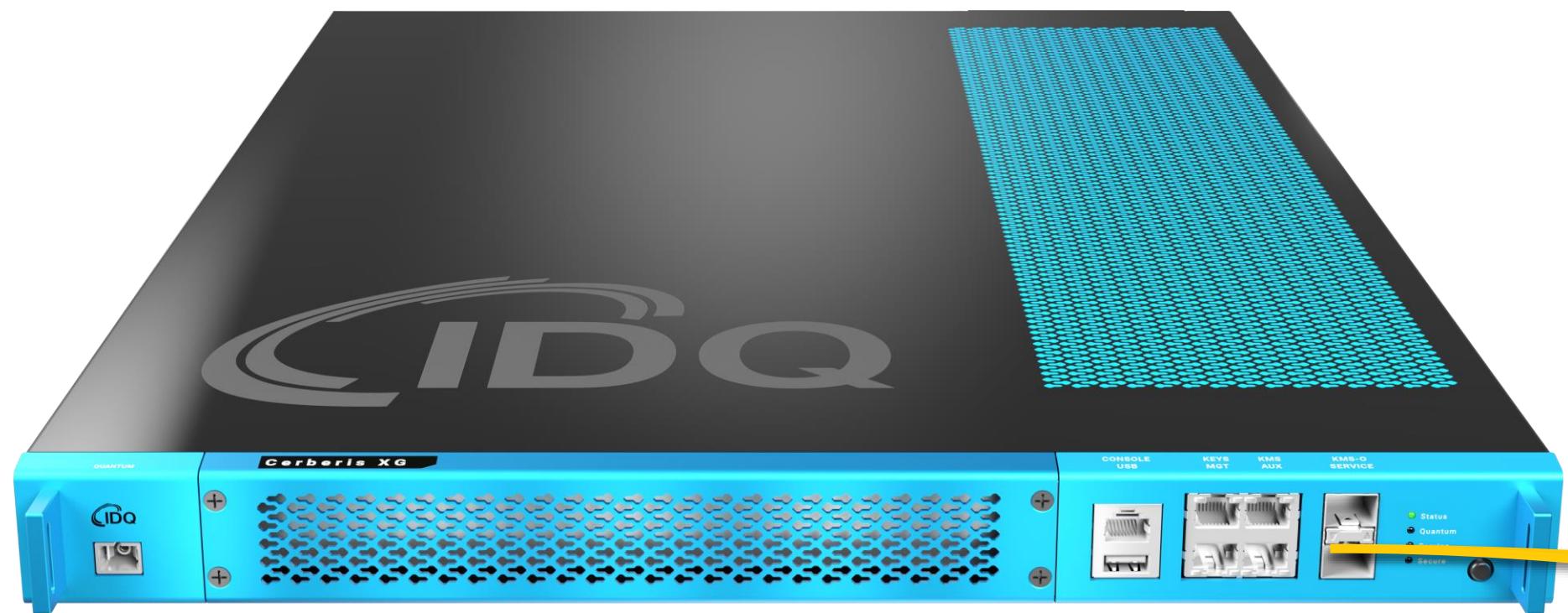


LC/UPC Attenuator  
(optional)

# Cerberis XG – Physical setup (Service Channel)

SWISS  
QUANTUM+

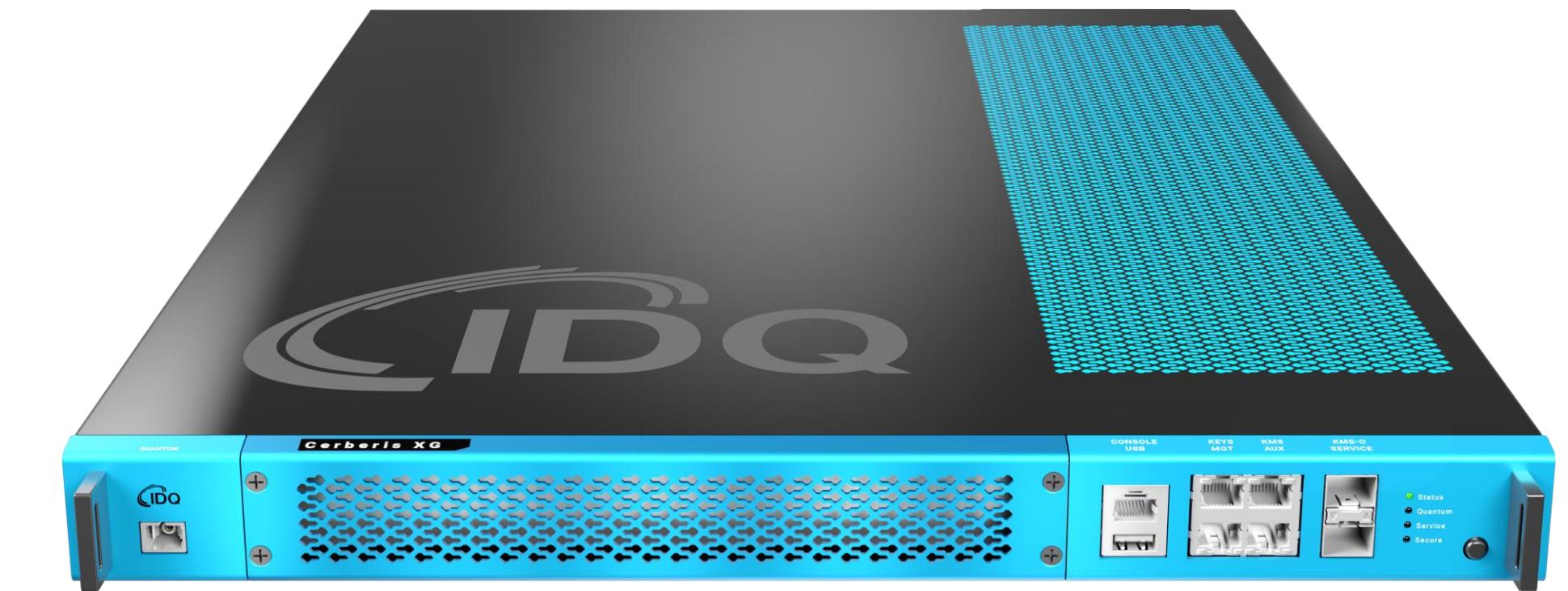
## Connect Service Channel



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

The valid RX power is between -9dBm and -28dBm  
(TX power is approximately +4dBm)

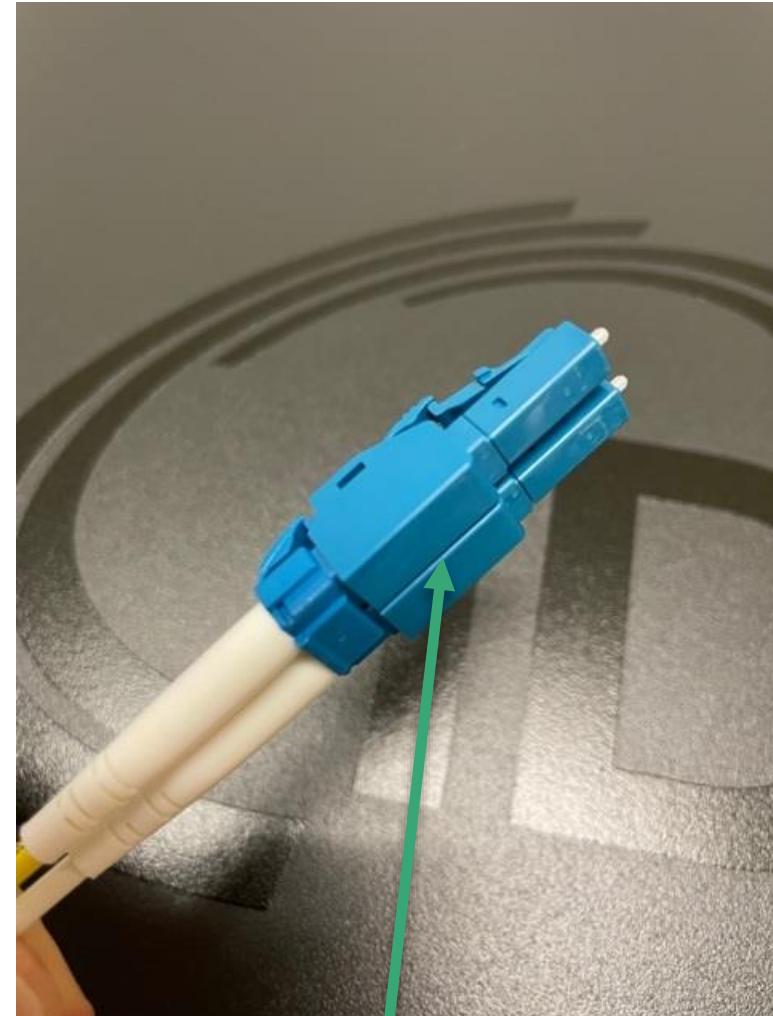
Be sure to measure the RX power to determine necessary losses



LC/UPC Attenuator  
(optional)

# Cerberis XG – Physical setup (Service Channel)

## Connect Service Channel



Attenuators



No Attenuators



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

# Physical Setup – Service Channel

## A.Optical Connections, Service Channel:

- 1.Remove both protective caps from the LC connector of the dual fiber patch cord and clean the exposed fiber ferrules by wiping them on the cleaning tool. Only wipe in the direction depicted on the cleaning tool!
- 2.If the service channel fibers have less than 10 dB of losses, add the provided fix LC attenuators to have an attenuation between 10 to 20dB to prevent saturation of the SFP modules.
  - i.Remove the protective caps on both sides of the LC fix attenuator and clean the exposed fiber ferrules by wiping them on the cleaning tool. Clean the interior of the LC fix attenuator with the provided cleaning tool.
  - ii.Connect the LC/UPC fiber connectors to the fix attenuators.

**IMPORTANT:** A proper connection is confirmed by an audible click when inserting the module.

- 3.Remove the protective cap from the SFP module and carefully insert the LC connector into the SFP module.
- 4.Slide the SFP module into the front panel mount at the QKD terminal in the slot labelled as Service. A proper installation is confirmed by an audible click when inserting the module.

**IMPORTANT:** A proper connection is confirmed by an audible click when inserting the module. Verify that the metallic handle is in a downward position!

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

## Details

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

# Installation procedure – Power ON

## LED Representation

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



# Cerberis XG – Console cables

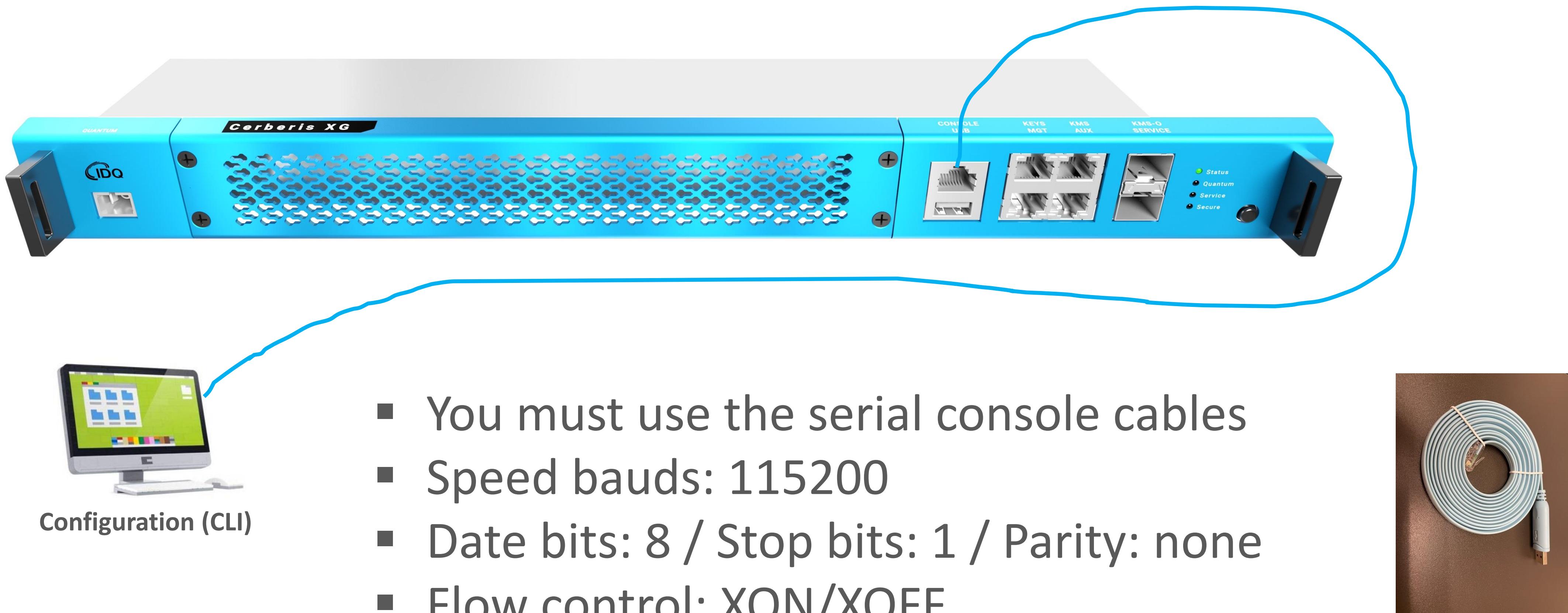
The QKD is delivered with console cables.

**They are mandatory** to do the basic configuration.



# Cerberis XG – Physical setup (Console)

- Preliminary configuration done with the CLI console port
- Alternatively use SSH to the default IP addresses



# CERBERIS XG

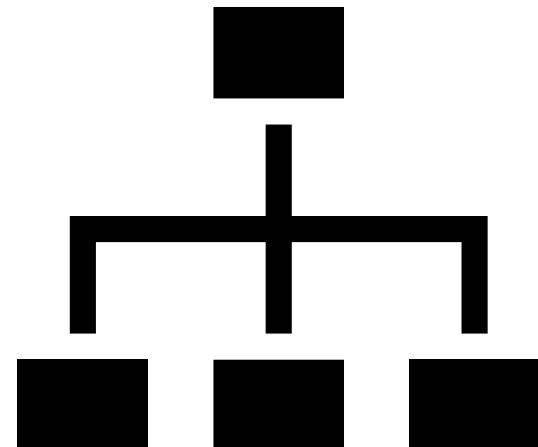
## Preliminary configuration

## 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 QKD system



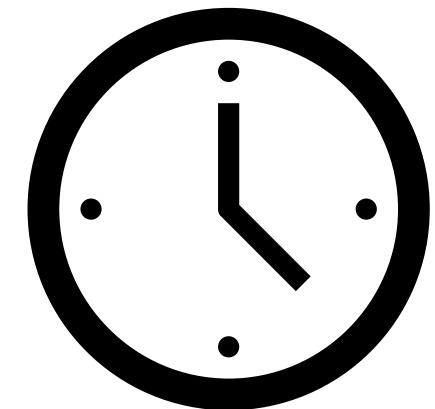
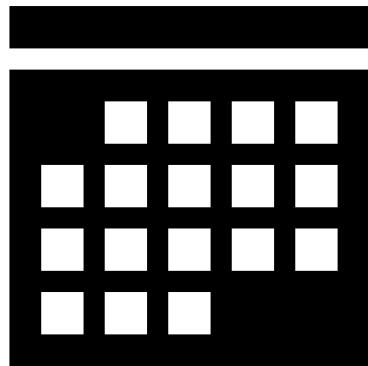
## Management:



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

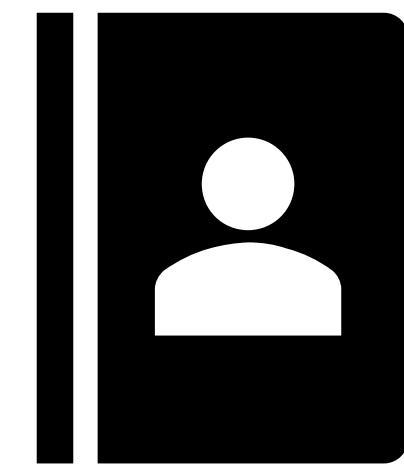
## ▶ Date & Time

- ⑩ It is necessary to configure the date and time for proper topology synchronization and network certificate use
- ⑩ Consider configuring an NTP server



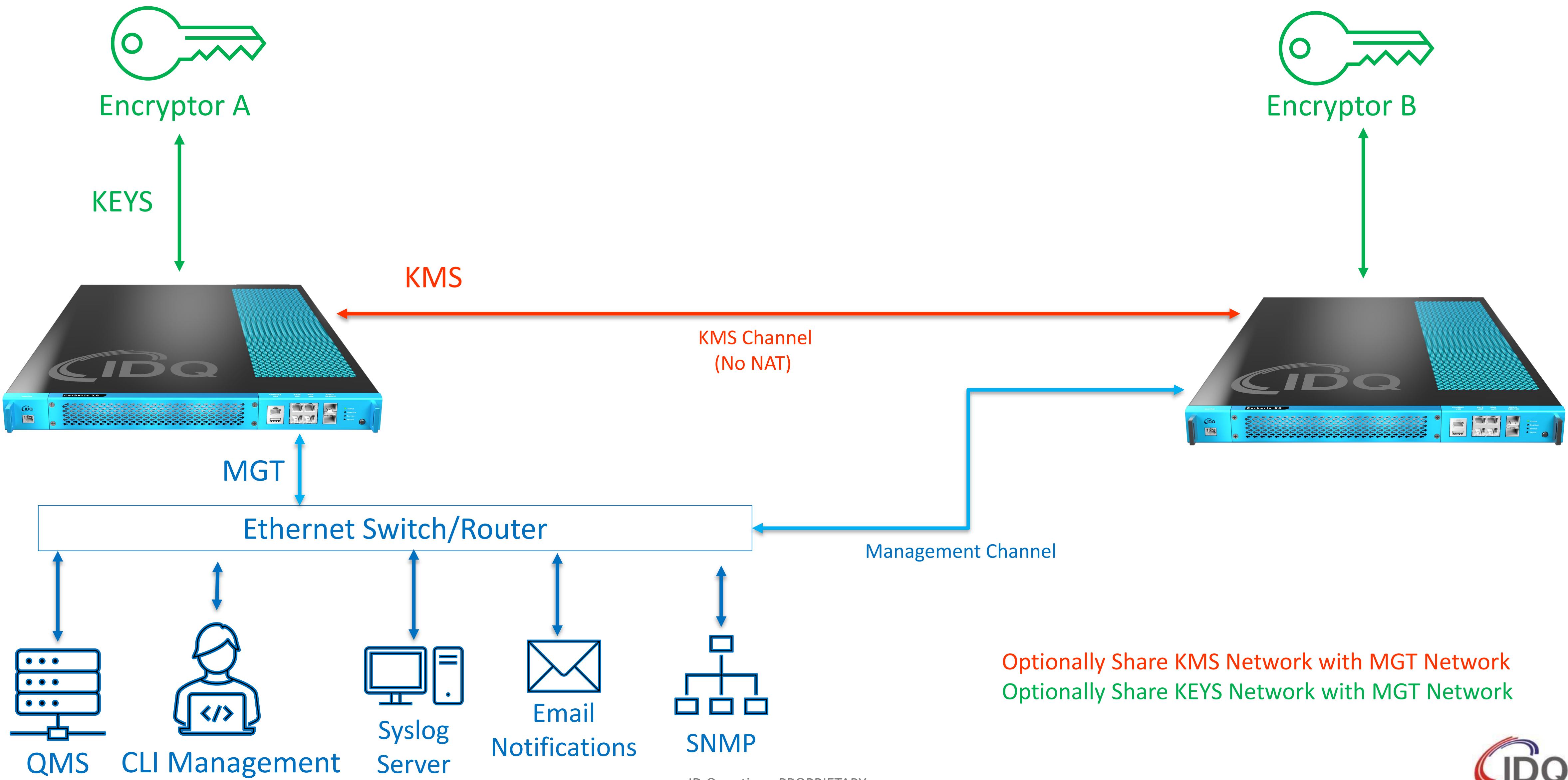
## ▶ IP Address Plan

- ⑩ Understand the necessary IP addresses
- ⑩ Choose the necessary IP addresses
- ⑩ Configure the MGT IP address
- ⑩ Configure the KMS IP address
- ⑩ Configure the KEYS IP address



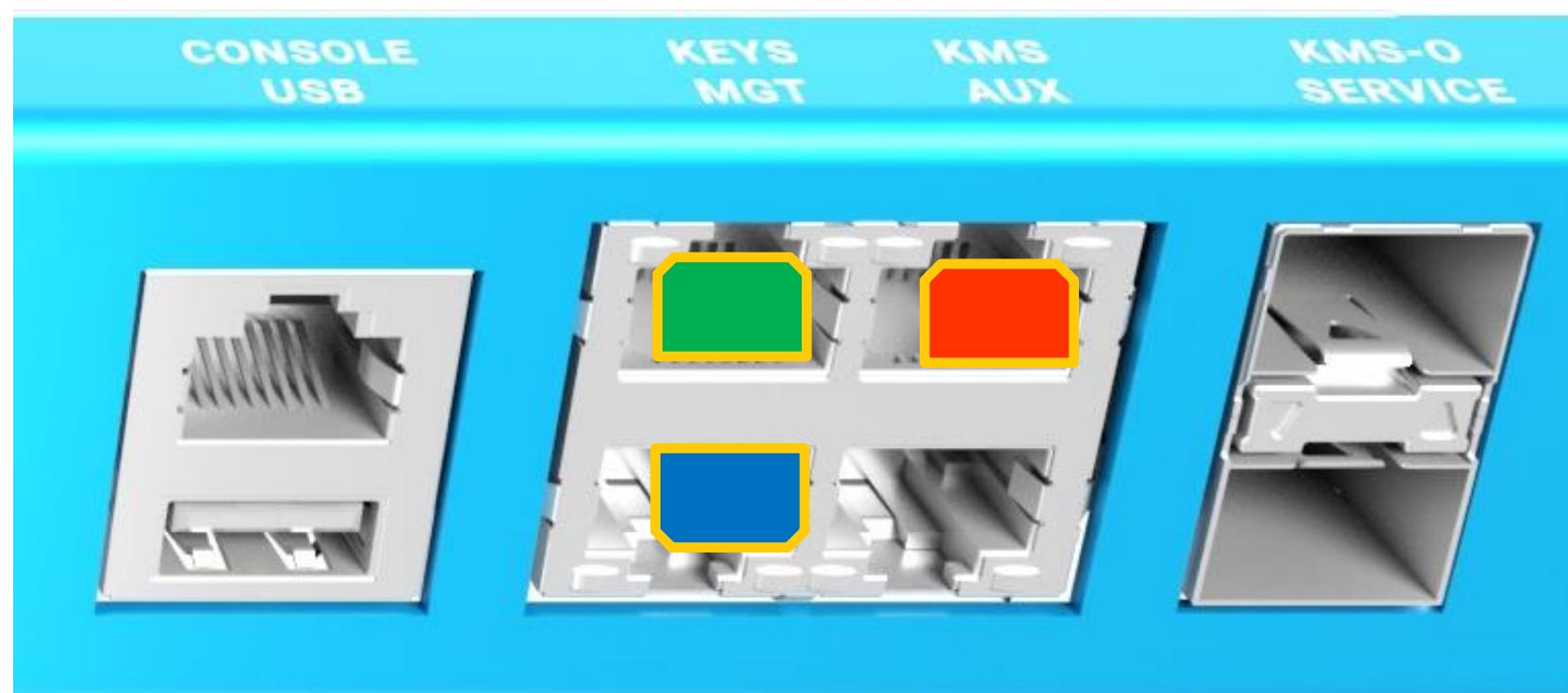
**X.X.X.X**

# IP Addressing Considerations



# IP Addressing Considerations

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



## LAB IP addresses Example

10.10.1.xxx QMS SERVER

10.10.1.xxx MGT ALICE

192.168.xxx.xxx KEYS ALICE

10.10.2.xxx KMS ALICE

10.10.3.xxx AUX ALICE

10.10.1.xxx MGT BOB

192.168.xxx.xxx KEYS BOB

10.10.2.xxx KMS BOB

10.10.3.xxx AUX BOB

Mask 255.255.xxx.xxx

Gateway 10.10.xxx.xxx

## LAB IP addresses Example

10.10.xxx.xxx QMS SERVER

10.10.xxx.xxx QNC ALICE

10.10.xxx.xxx QKD ALICE

10.10.xxx.xxx Shelf Manager Alice bound (used for upgrade)

10.10.xxx.xxx Switch Management Alice (used for management)

192.168.xxx.xxx ETSI frontpane p2 ALICE

10.10.xxx.xxx QNC BOB

10.10.xxx.xxx QKD BOB

10.10.xxx.xxx Shelf Manager BOB bound (used for upgrade)

10.10.xxx.xxx Switch Management BOB (used for management)

192.168.xxx.xxx ETSI frontpane p2 BOB

Mask 255.255.xxx.xxx

Gateway 10.10.xxx.xxx

# Cerberis XG – Preliminary configuration

Connect to the CerberisXG via the CONSOLE Port

Alternatively, SSH to the device using the Default IP address:

192.168.10.102 (Alice)

192.168.10.107 (Bob)

Default Credentials:

Username: admin

Password: admin

Type

“date --help” to change the date/time

Change the time zone before changing the time

(Alternatively consider configuring NTP server)

Type

“network -i 0” to change the MGT IP address

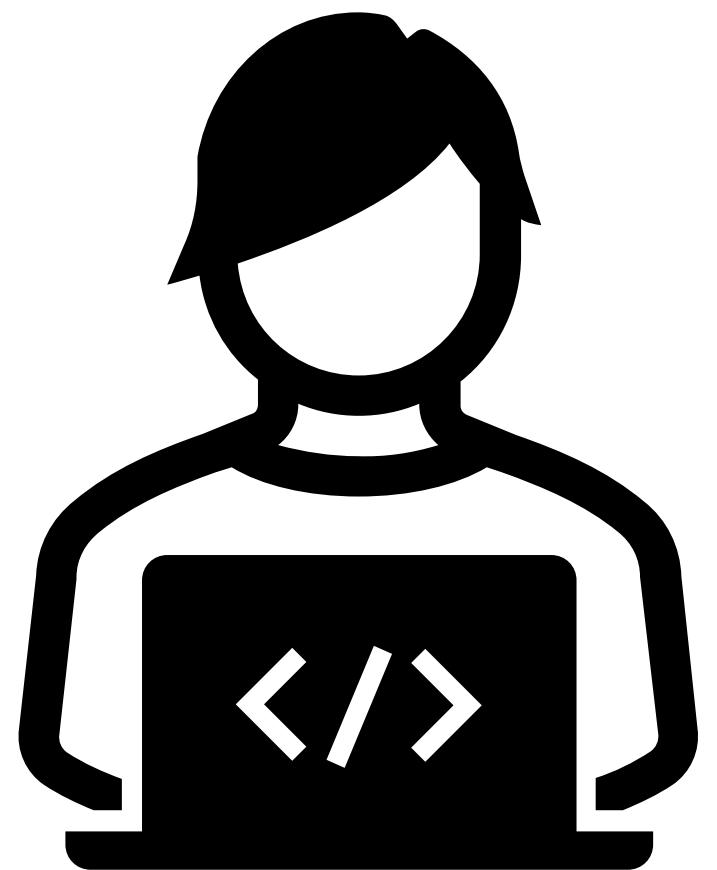
Mandatory

“network -i 1” to change the KEYS IP address

Mandatory... even if you are  
not using this AUX interface!

“network -i 2” to change the AUX IP address

“network -i 3” to change the KMS IP address



# Cerberis XG - Initial Login



1. Login as “admin”:

> admin

default password: admin

2. The first time the system is accessed as admin a password change is requested.

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

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

# Cerberis XG – Preliminary configuration

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

Usage: date [command] [options]

Options:
  --help Show help information.

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

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

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

Help

Command to configure timezone

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

Usage: date set [options] <datetime>

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

Options:
  --help     Show help information.

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

Help

Command to configure date and time

# Cerberis XG – Preliminary configuration

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

Usage: network [command] [options]

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

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

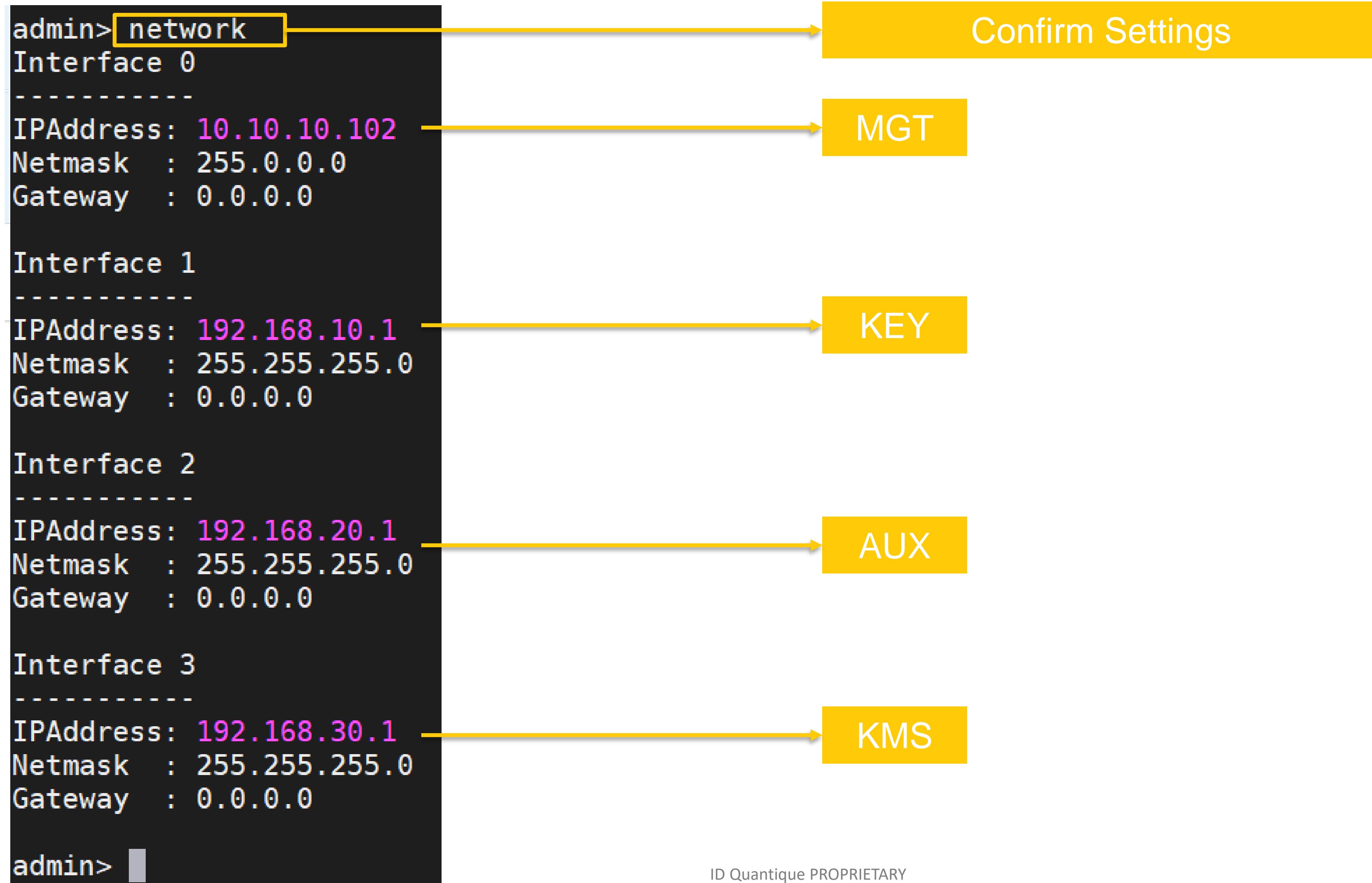
Run 'network [command] --help' for more information about a command.

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

Help

Set CerberisXG/QNC  
management IP address

# Cerberis XG – Preliminary configuration



# Cerberis3 – Preliminary configuration



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

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

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

admin>
```

Confirm Settings

Optional QNC Frontpane Key Delivery  
Interfaces

# Cerberis 3 – Preliminary configuration

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

Usage: network [command] [options]

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

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

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

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

Configure the Frontpane port using the “network” command:

Help and command's information

Command to use with arguments



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

# Cerberis 3 – Preliminary configuration

```
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

# QNET & QMS installation



## Migration from a Previous Version:

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

Export the current KEMS/QMS configuration using QNET:

```
qnet export group -o <filename> <group name>
```

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

```
sudo docker-compose down
```

(Optionally destroy all old Docker containers using “sudo docker system prune” / “sudo docker system prune -a”)

Perform the QKD system and QNET/QMS software upgrade procedure

Import the old configuration using:

```
qnet import group <group name> <file name>
```

### ALTERNATIVELY:

The current QMS supports auto-migration from one QMS version to the next.

Copy the new QMS tar file to the same directory which contains the QMS folder.

Decompress the new QMS package as per the installation instructions. It should maintain the existing configuration.



## Docker installation

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



## Pre-requisite

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

Uninstall older versions of Docker and Docker-Compose

# Docker Installation



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

(typically something like “`sudo apt install docker`”

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

```
# sudo systemctl start docker
```

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

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

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

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

# Docker-Compose Installation



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

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

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

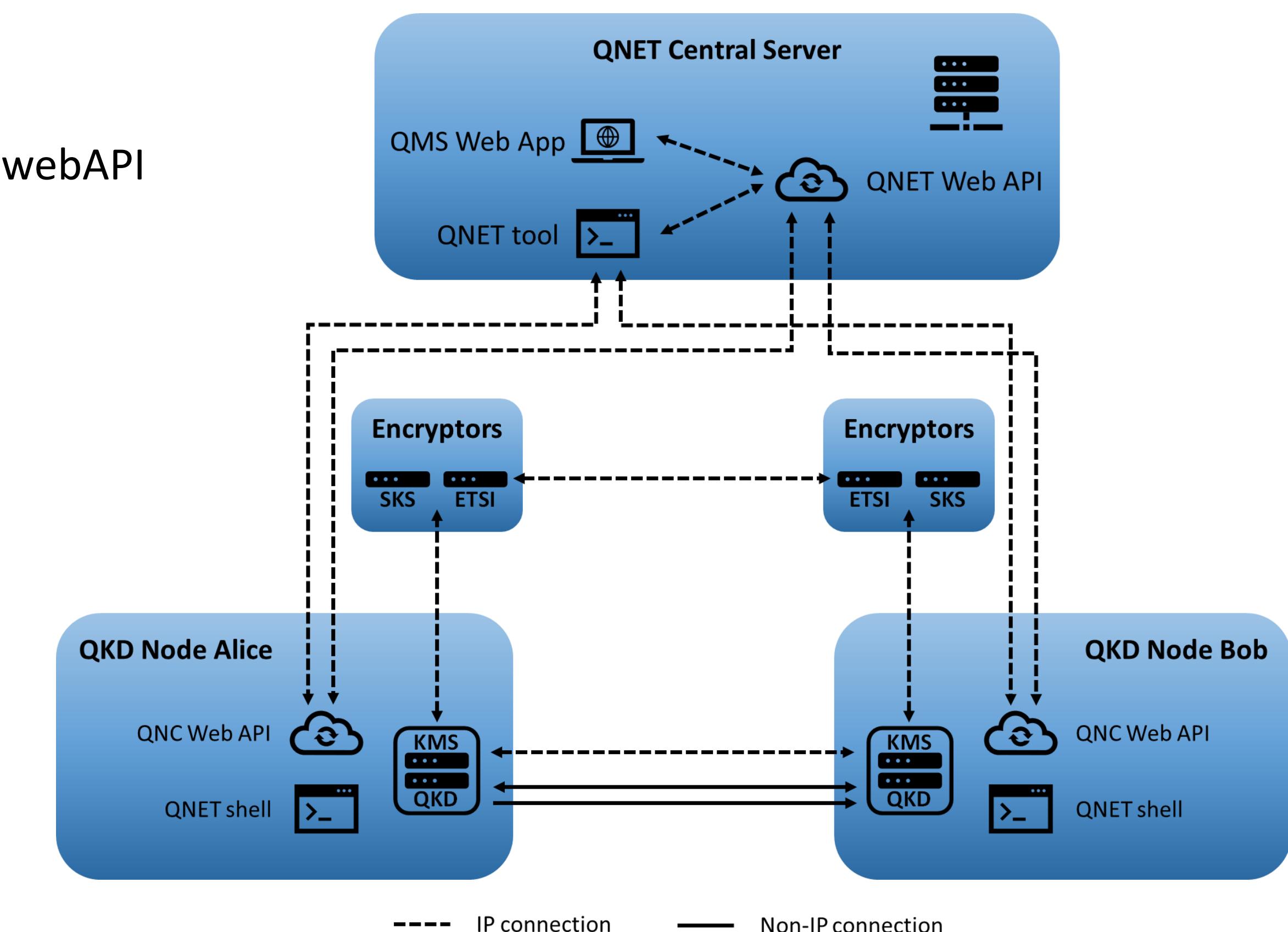
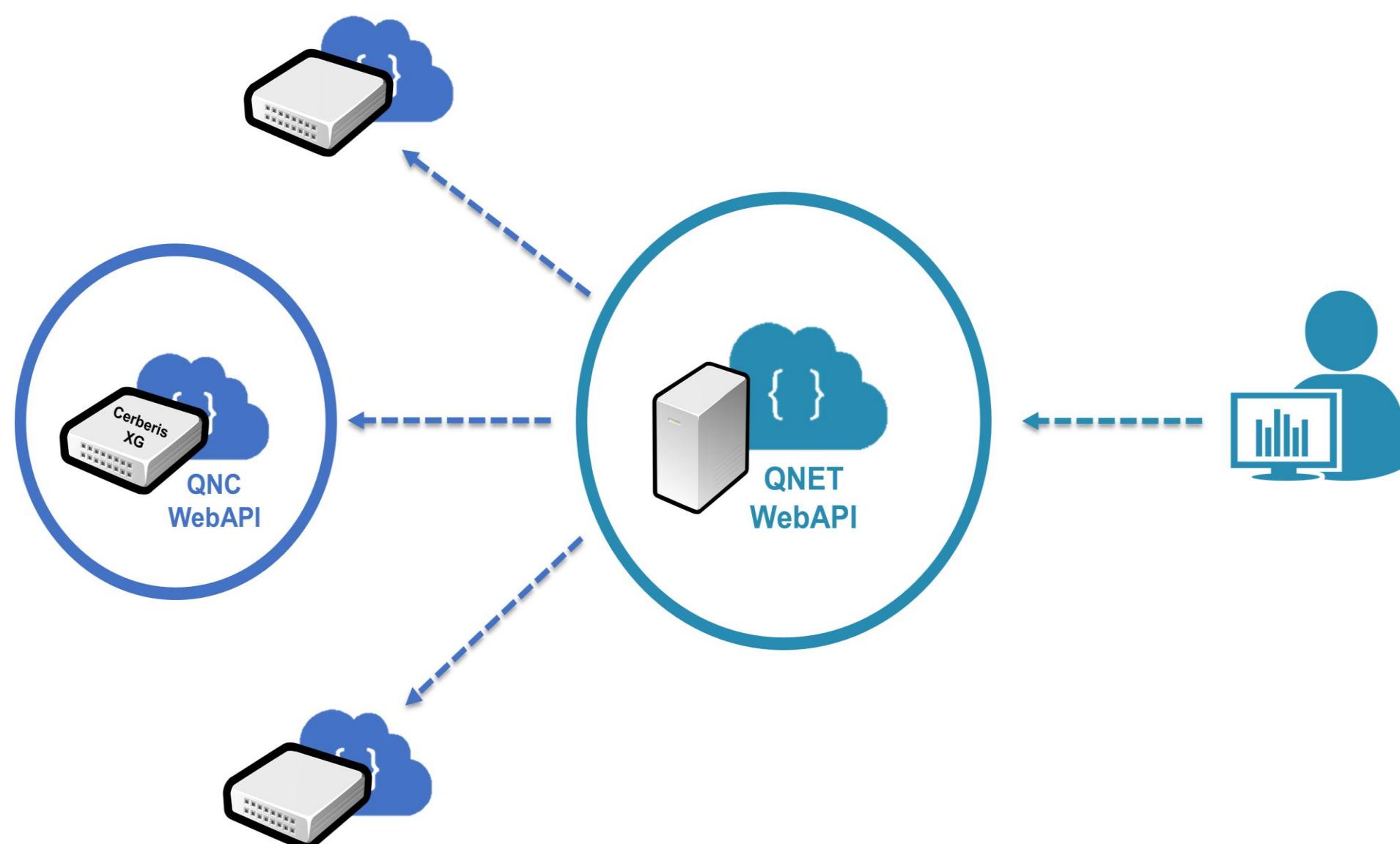
For more information, visit:

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

## Installing QNET Central Server for configuring and monitoring

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

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



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

# QNET Installation

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

```
chris@chris-VirtualBox:/home/idq$ sudo ./qnet-installer-1.1.74.24404-20210715-linux-x64.run
Verifying archive integrity... 100% MD5 checksums are OK. All good.
Uncompressing Installer for QNET, an administration tool for quantum networks 100%
=====
[REDACTED]
=====
For qnet version 1.1.74
=====

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

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

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

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

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

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

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

## QMS installation

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

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

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

Note you may need to use sudo permissions

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

Use this command to load the images into docker:

***sudo docker load -i images.tar***

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

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

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

***cp customization/.env .***

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

# QMS Installation

Launch the service with this command:

**docker-compose up -d**

You can stop the QMS services with this command:

**docker-compose down**

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

## Docker installation

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

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

- To check if the containers started

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

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
dc5f2a3942dc	idq/qms-backend:1-0-2	"dotnet Ems.Platform..."	2 hours ago	Up 2 hours		qnet-package-qms-115_qms-backend_1
1d0c10380aa0	idq/qms-postgres:10.17	"docker-entrypoint.s..."	2 hours ago	Up 2 hours	5432/tcp	qnet-package-qms-115_qms-postgres_1
ebefbf7d28fec	idq/qnet-oauth:v1.1.74	"/bin/sh -c 'echo \"T...\"	2 hours ago	Up 2 hours	80/tcp, 443/tcp, 0.0.0.0:8143->8143/tcp	qnet-package-qms-115_qnet-oauth_1
6f7ea643ed68	idq/qnet-mariadb:v1.1.74	"docker-entrypoint.s..."	2 hours ago	Up 2 hours	3306/tcp	qnet-package-qms-115_qnet-mariadb_1
1f47a797d2ff	idq/qms-mongo:4.0.24	"docker-entrypoint.s..."	2 hours ago	Up 2 hours	27017/tcp	qnet-package-qms-115_qms-mongo_1
0776fb3b8e1	idq/qms-frontend:1-0-2	"/docker-entrypoint..."	2 hours ago	Up 2 hours	0.0.0.0:6060->6060/tcp, 80/tcp, 0.0.0.0:6443->6443/tcp	qnet-package-qms-115_qms-frontend_1
7d1f924dd85a	idq/qnet-webapi:v1.1.74	"/bin/sh -c 'echo \"T...\"	2 hours ago	Up 2 hours	80/tcp, 0.0.0.0:443->443/tcp	qnet-package-qms-115_qnet-webapi_1
6be666e5a7a9	idq/kems_database:v2-1-35	"docker-entrypoint.s..."	3 weeks ago	Up 2 hours	3306/tcp	kems-config-api-v2-1-35_kems-database_1

- To check if the containers started

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

## Docker installation

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

- **disable SE Linux**

```
# vi /etc/selinux/config
```

...

```
SELINUX=disabled
```

...

- **disable the firewall**

```
# systemctl stop firewalld
```

```
# systemctl disable firewalld
```

# Sample QNet Commands (Script):



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

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

qnet create node CerberisXGA Group1 -type Mgmt -ip 10.10.10.102 -uid cf8658d9b... -id550c5a7360bb1
qnet create node-network-interface CerberisXGA Key 192.168.10.1
qnet create node-network-interface CerberisXGA Kms 192.168.30.1

qnet create node CerberisXGB Group1 -type Mgmt -ip 10.10.10.107 -uid 0a210554d2c... -id8ce31da1f70
qnet create node-network-interface CerberisXGB Key 192.168.10.2
qnet create node-network-interface CerberisXGB Kms 192.168.30.2

qnet create node-link KMSLINK CerberisXGA CerberisXGB

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

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

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

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

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

qnet create path CiscoA CiscoB -mode Automatic
qnet create path ETSIA ETSIB -mode Automatic

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

qnet deploy group Group1

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

qnet deploy group Group1
```

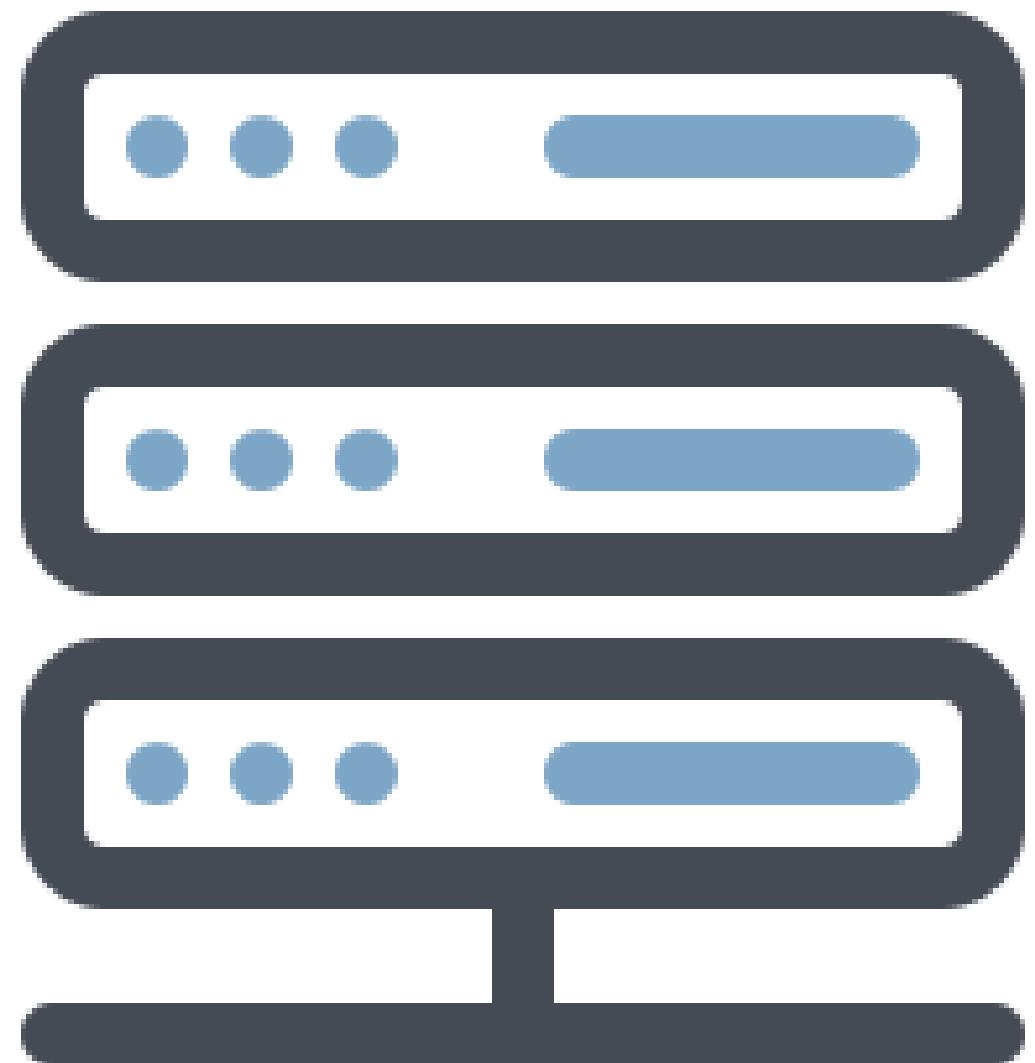


Before Moving Forward with QMS Configuration, It may be Quicker/Easier to use a QNET Configuration Script Instead:

The following slides will detail both options.  
(Note you will need to configure Node name in CLI using “qnc --set-name” command)

Note that Clavis3 uses provider type “Cerberis3” and not “Clavis300”.

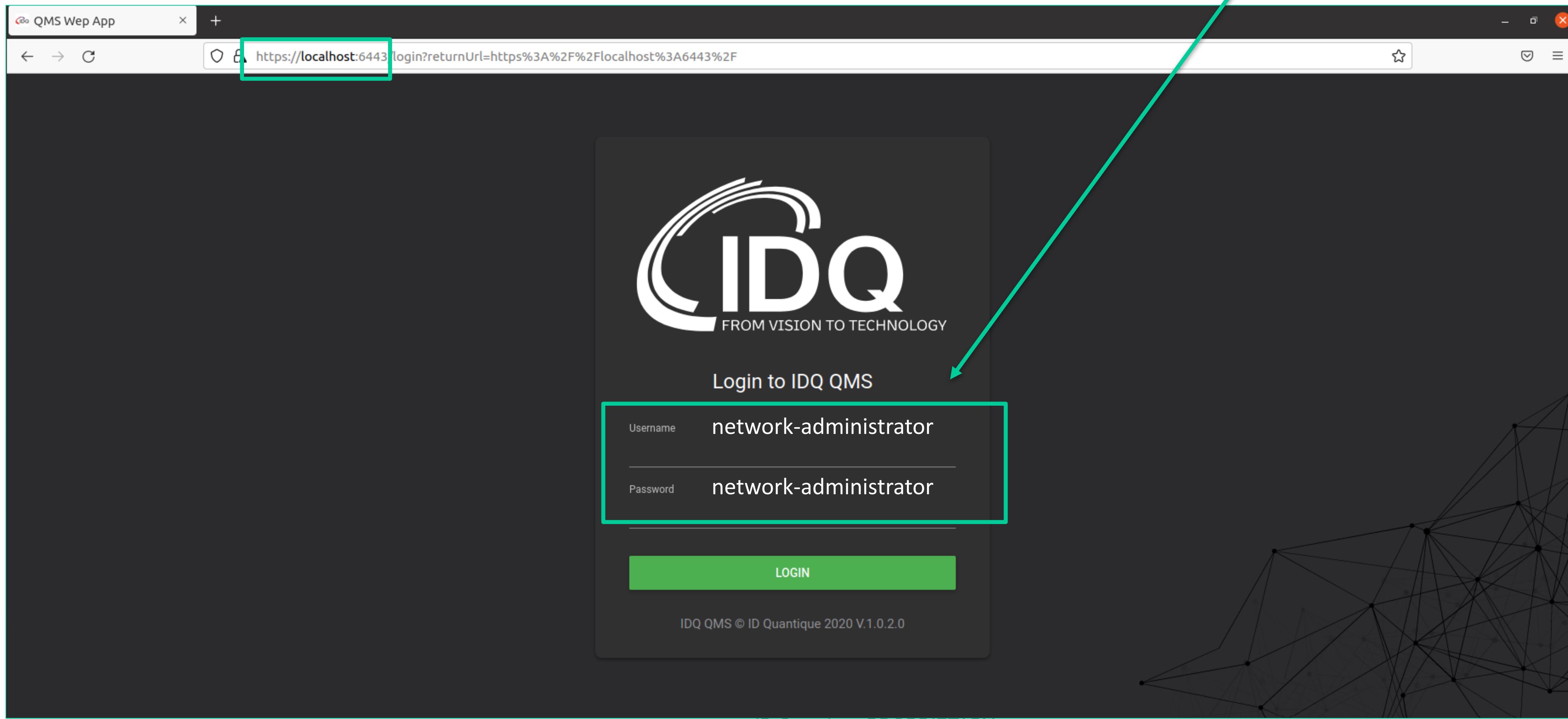
# QMS configuration



# QMS installation

## Access to the GUI

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



The screenshot shows the QMS Web App interface with several highlighted features:

- URL of QMS Tool**: Points to the browser address bar showing `https://localhost:6443`.
- Configurator Position**: Points to the "Dashboard" button in the top navigation.
- Contextual Help Toggle**: Points to the question mark icon in the top right corner.
- Language**: Points to the language selection dropdown set to English.
- User/Logout**: Points to the user account icon in the top right corner.
- Navigation Pane**: Points to the left sidebar containing links for Dashboard, Nodes, Paths, Monitoring, Firmware, Settings, and Users.
- QKD Node Hierarchy Configuration**: Points to the tree view under the "BostonLab" section, which includes branches for QNCA, QNCB, ETSIA, QKDA, ETSIB, and QKDB, each with edit icons.
- Node Positions on any map you choose**: Points to a map of Europe with two nodes plotted.
- Events (Below)**: Points to the event log table below the map.
- Auto Refresh**: Points to the "Auto Refresh" toggle switch at the bottom of the event log.
- Last Updated: 2:02:12 PM**: Points to the timestamp next to the auto refresh switch.

The dashboard also displays the following information:

- Quantum node**: The main title of the dashboard.
- Search**: A search bar.
- BostonLab**: A collapsed section in the hierarchy.
- Events**: A section for viewing event logs.
- V.1.0.2.0**: The software version number.

# Monitoring



## Known Limitations

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

## Naming on QMS

- The naming is different than those seen before
  - KMS → “CerberisXG” or “Cerberis3 QNC” Acting as KMS for this site
  - Provider → “CerberisXG” or “Cerberis3 QKD”
  - Consumer → Encryptor
  - KMS link → Link between “CerberisXG” KMS or “Cerberis3 QNC” KMS
  - Provider link → Link between “CerberisXG” or “Cerberis3 QKD”
  - Consumer link → Link between encryptors

## Preliminary configuration

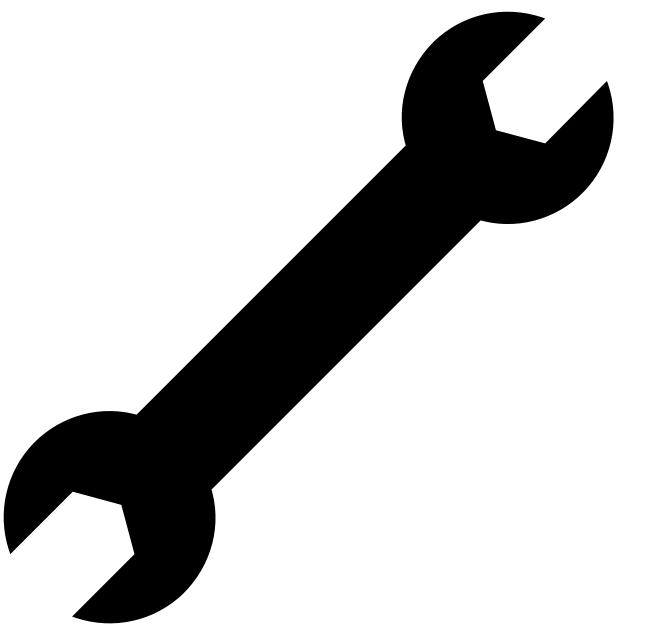
- Set the QNC name on each CerberisXG or Cerberis3 QNC and copy the UID

## QMS Configuration

- Group
  - Create QMS Group
- Node
  - Create a Node for each QKD device
  - Create a Node link for each QKD pair
- Provider
  - Create a Provider for each QKD device
  - Create a Provider link for each QDK pair

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

# Prepare Node



# Prepare Node/KMS



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

***qnc --set-name <system\_name>***

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

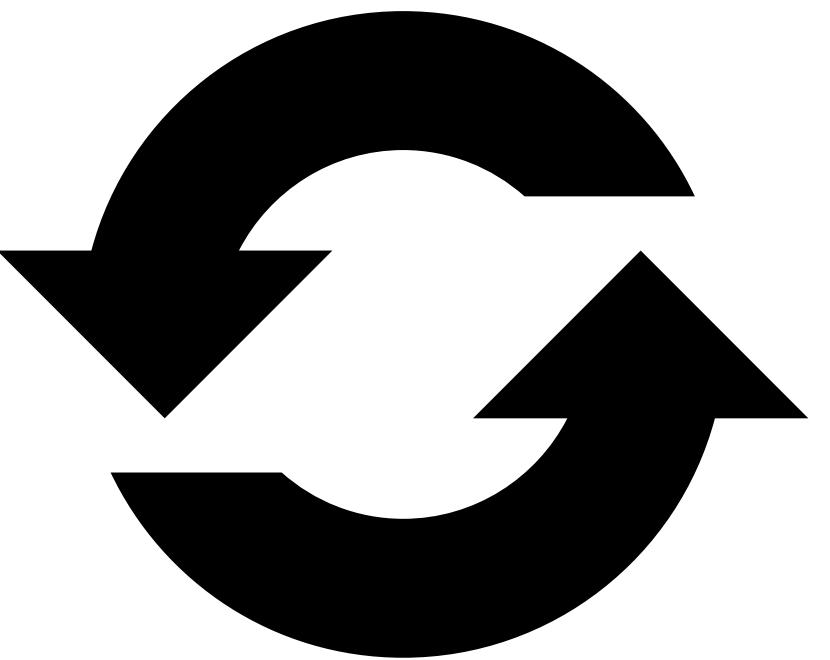
- Execute the following command to check the name of the KMS:

**qnc**

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

- Write down the **UID**

Repeat the same process to prepare the peer node



# Prepare Node/KMS



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

***qnc --set-name <system\_name>***

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

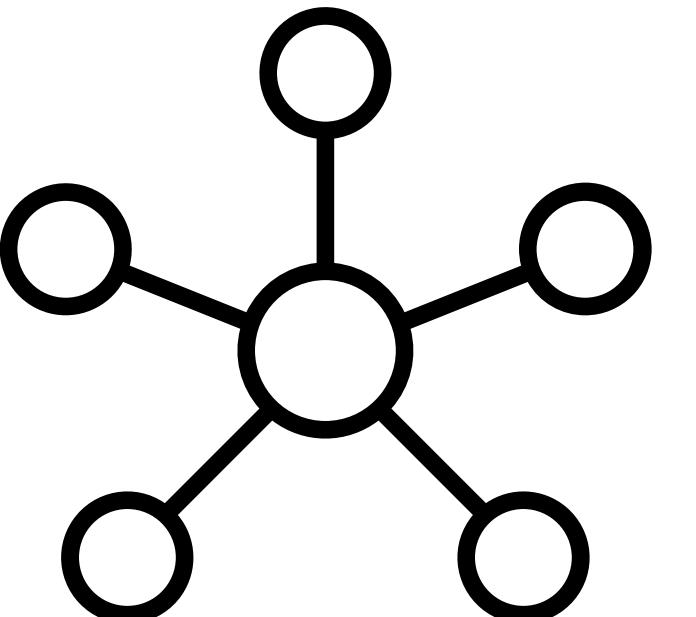
- Execute the following command to check the name of the KMS:

**qnc**

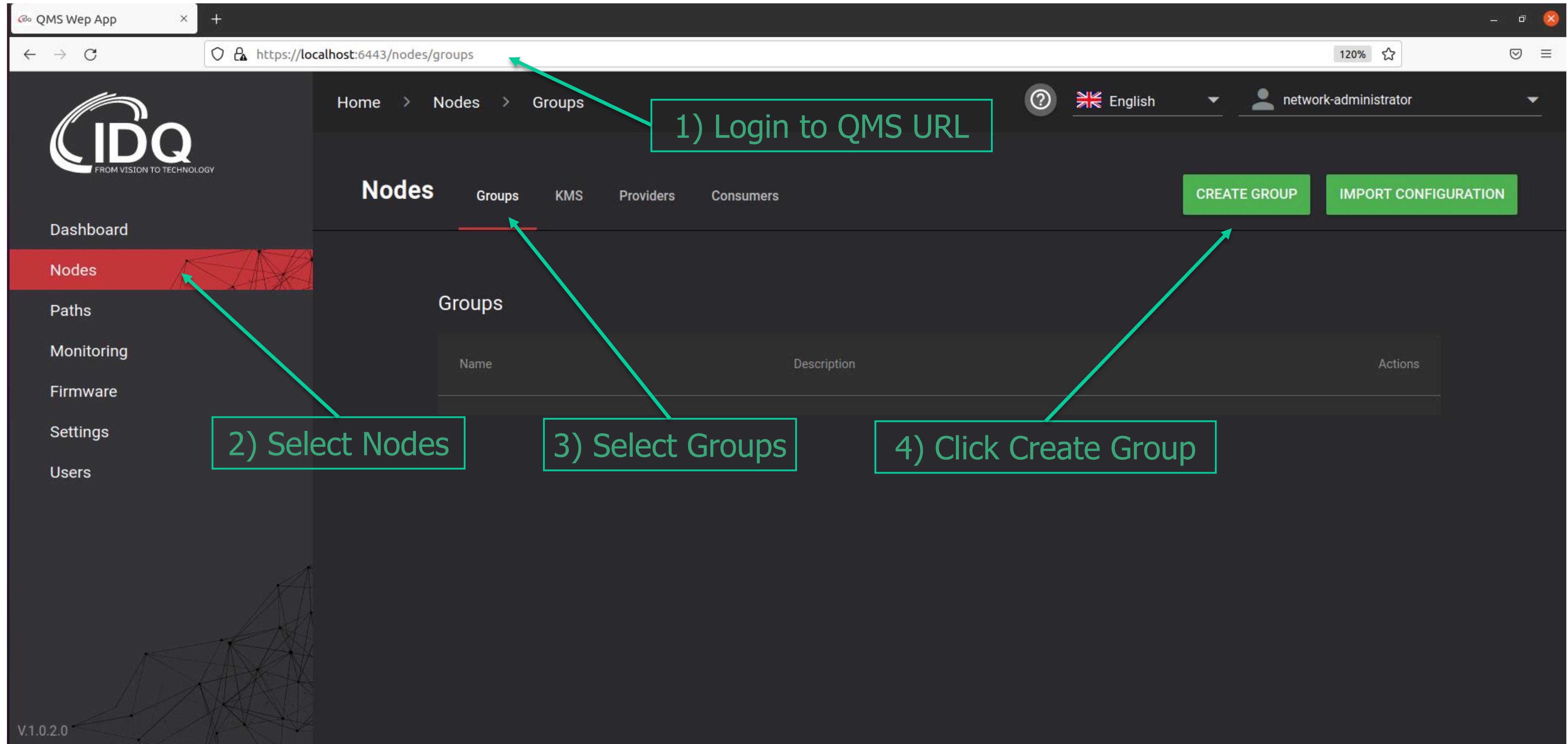
```
admin> qnc
Product: QNC
QNC Name: QNCB
QNC UID: 0a210554d2c84dfdbfd9e8ce31da1f70
Firmware: 0.9.0-24182-QNC
Serial Number: 42180051
admin>
```

- Write down the **UID**

# Create Group



# Create Group



# Create Group

SWISS  
QUANTUM+

The screenshot shows a web application interface for creating a group. The left sidebar has 'Nodes' selected. The main area is titled 'Create group step 1/1'. It contains an 'Informations' section with 'Name' set to 'BostonLab' and 'Description' set to 'Boston Test Lab'. A large green 'CREATE' button is at the bottom. Two teal arrows point from boxes labeled '1) Configure Name and Description' and '2) Click Create' to the respective 'Name' field and 'CREATE' button.

QMS Wep App

https://localhost:6443/nodes/groups

-administrator

Dashboard

Nodes

Paths

Monitoring

Firmware

Settings

Users

V.1.0.2.0

1) Configure Name and Description

2) Click Create

CREATE

# Create Group

SWISS  
QUANTUM+

The screenshot shows the QMS Web App interface for managing groups. The left sidebar has links for Dashboard, Nodes (which is selected and highlighted in red), Paths, Monitoring, Firmware, Settings, and Users. The main content area shows a navigation bar with Home > Nodes > Groups. Below this is a sub-navigation bar with Nodes (selected), Groups, KMS, Providers, and Consumers. On the far right are buttons for CREATE GROUP and IMPORT CONFIGURATION. The main table displays a single group entry:

Name	Description	Actions
BostonLab	Boston Test Lab	

Two green arrows originate from callout boxes at the bottom of the screen. One arrow points from a box labeled "Verify New Group" to the "BostonLab" row in the table. Another arrow points from a box labeled "Modify if Needed" to the edit icon in the "Actions" column of the same row.

# Create Nodes



# Create Nodes/KMS

The screenshot shows the QMS Web App interface at <https://localhost:6443/nodes/kms>. The left sidebar has 'Nodes' selected. The main navigation bar shows 'Nodes > KMS'. The 'KMS' tab is highlighted. The top right shows 'Groups: BostonLab'. A green arrow points from the 'Nodes' sidebar to the 'Nodes' tab in the navigation bar. Another green arrow points from the 'KMS' tab to the 'KMS' tab in the navigation bar. A third green arrow points from the 'CREATE NODE' button to the 'CREATE NODE' button in the top right.

1) Select Nodes

2) Select KMS

3) Click Create Node

# Create Nodes/KMS

QMS Web App

https://localhost:6443/nodes/kms

Node Creation step 1/2

Identification information

Name  
QNCA

KMS unique ID  
cfa8658d9b4744cab5d550c5a7360bb1

Select a group  
BostonLab

NEXT

1) Name must exactly match the QNC name configured in CLI

2) UID must exactly match the UID from the QKD system (copied earlier)

3) Select Group Membership

4) Click Next

# Create Nodes/KMS

QMS Wep App

https://localhost:6443/nodes/kms

administrator

Node Creation step 2/2

Interfaces

Type Mgmt

IP address 10.10.10.1

ADD NEW INTERFACE

2) Enter IP address of CerberisXG MGT port or Cerberis3 QNC

1) Select Interface Type:  
• "Mgmt" for CerberisXG using dedicated MGT port  
• "Any" For sharing CerberisXG MGT port for KMS and/or KEY traffic - or Cerberis3

3) Click Add New Interface

4) Do Not Click Create Yet!

PREVIOUS

CREATE

V1.0.2.0

# Create Nodes/KMS

QMS Wep App https://localhost:6443/nodes/kms 120% administrator

## Node Creation step 2/2

1) Create another interface for each in use:  
• “Key” for CerberisXG using KEY interface – or Cerberis3 using frontpane key interface

Interfaces

Type Key

IP address 192.168.10.1

ADD NEW INTERFACE

2) IP Address of CerberisXG KEY interface – or Cerberis3 Frontpane interface

3) Click Add New Interface

4) Do Not Click Create Yet!

PREVIOUS

CREATE

V.1.0.2.0

# Create Nodes/KMS

SWISS  
QUANTUM+

QMS Wep App https://localhost:6443/nodes/kms 120% administrator PORT CONFIGURATION

Node Creation step 2/2

1) Create another interface for each in use:  
• “Kms” for CerberisXG using KMS interface – Not used for Cerberis3

Interfaces

Type	IP address	Mgmt	Key
Kms	192.168.30.1	10.10.10.1	192.168.10.1
<b>ADD NEW INTERFACE</b>			

2) IP Address of CerberisXG KMS interface

3) Click Add New Interface

4) Do Not Click Create Yet!

PREVIOUS

CREATE

V.1.0.2.0

# Create Nodes/KMS

**Node Creation** step 2/2

Interfaces

Type	IP		
Mgmt	10.10.10.1		
Key	192.168.10.1		
Kms	192.168.30.1		

1) Verify all the needed interfaces have been added

2) Click Create

PREVIOUS

CREATE

# Create Nodes/KMS

SWISS  
QUANTUM+

QMS Wep App X +

https://localhost:6443/nodes/kms 120% ⭐

IDQ FROM VISION TO TECHNOLOGY

Dashboard

Nodes

Paths

Monitoring

Firmware

Settings

Users

V.1.0.2.0

## Node Created

You can now choose among the following actions

- Nodes
- Edit Node
- Create another node

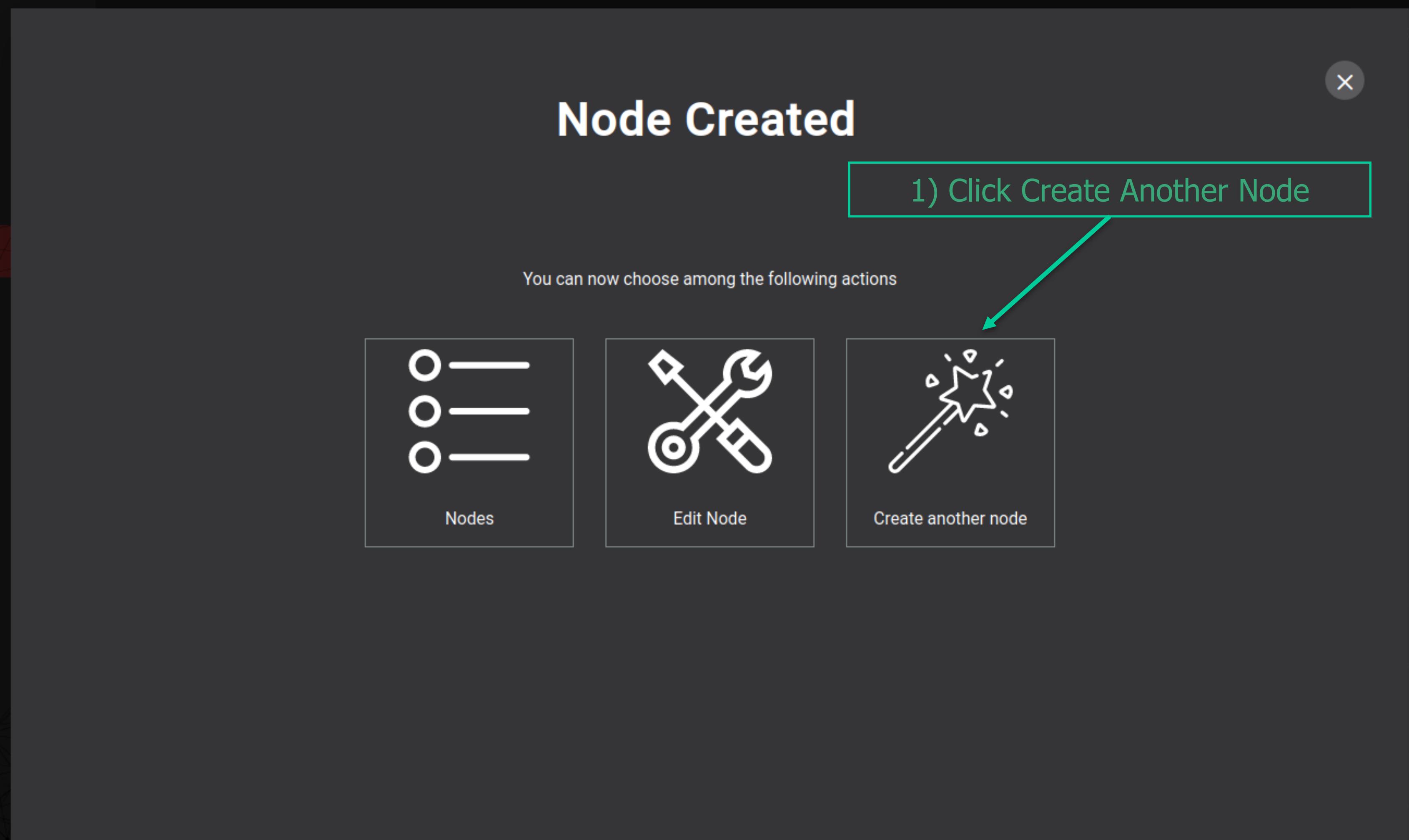
1) Click Create Another Node

-administrator

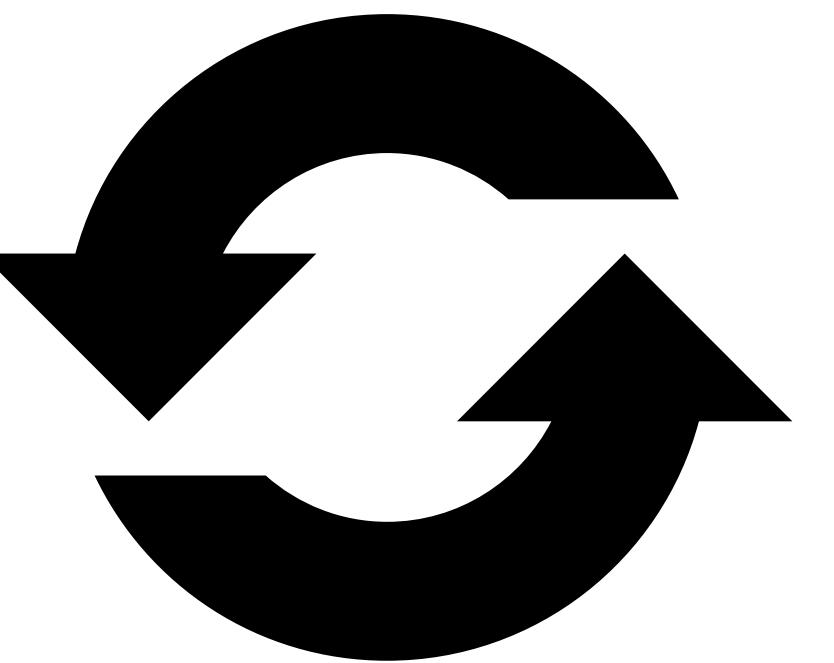
PORT CONFIGURATION

Actions

Pages (0 items)



Repeat the same process to create the peer node



# Create Nodes/KMS

Node Creation step 1/2

Identification information

Name: QNCB

KMS unique ID: 0a210554d2c84dfdbf9e8ce31da1f70

Select a group: BostonLab

NEXT

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

Node Creation step 2/2

Interfaces

Type	IP
Mgmt	10.10.10.2
Key	192.168.10.2
Kms	192.168.30.2

ADD NEW INTERFACE

PREVIOUS CREATE

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

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

# Create Nodes/KMS

QMS Wep App X +

https://localhost:6443/nodes/kms 120% ⭐

Dashboard

Nodes

Paths

Monitoring

Firmware

Settings

Users

V.1.0.2.0

## Node Created

You can now choose among the following actions

1) Nodes

Nodes

Edit Node

Create another node

PORT CONFIGURATION

administrator

Actions

ages (1 items)

The screenshot shows a web application interface for managing nodes. On the left, there's a sidebar with navigation links: Dashboard, Nodes (which is highlighted in red), Paths, Monitoring, Firmware, Settings, and Users. Below the sidebar, the version number V.1.0.2.0 is displayed. The main content area has a dark background. At the top, it says "Node Created". Below that, there's a message: "You can now choose among the following actions". Three buttons are shown: "Nodes" (with three circles icon), "Edit Node" (with crossed wrenches icon), and "Create another node" (with a star and gear icon). A green callout box labeled "1) Nodes" has a green arrow pointing to the "Nodes" button. On the right side of the screen, there's a sidebar titled "PORT CONFIGURATION" with a user "administrator" logged in. The bottom right corner features the ID Quantique logo.

# Create Nodes/KMS

SWISS  
QUANTUM+

QMS Wep App

https://localhost:6443/nodes

Home > Nodes > KMS

English network-administrator

Groups: BostonLab

CREATE NODE IMPORT CONFIGURATION

Nodes Groups KMS Providers Consumers

Dashboard

Nodes Paths Monitoring Firmware Settings Users

EXCEL EXPORT

Name Group name Internal Broker port Exchange port Administration port KemsProtocol port Actions

Name	Group name	Internal Broker port	Exchange port	Administration port	KemsProtocol port	Actions
QNCA	BostonLab	9000	9001	9002	9004	
QNCB	BostonLab	9000	9001	9002	9004	

Items per page: 10

1 of 1 pages (2 items)

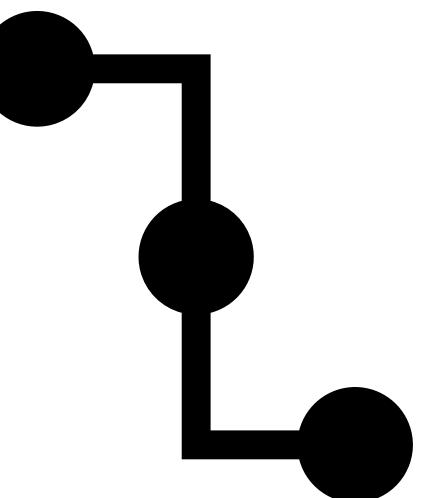
V.1.0.2.0

Verify your pair of nodes

Edit if needed

Advanced: Update Node position on map if desired

# Create Node KMS Links



# Create Node Links

QMS Wep App https://localhost:6443/nodes/kms 140% network-administrator

Home > Nodes > KMS

English

IDQ FROM VISION TO TECHNOLOGY

Groups BostonLab

Nodes Groups KMS Providers Consumers

CREATE NODE IMPORT CONFIGURATION

1) Select Nodes 2) Select KMS 3) Select View

EXCEL EXPORT

Name	Group name	Internal Broker port	Exchange port	Administration port	KemsProtocol port	Actions
QNCA	BostonLab	9000	9001	9002	9004	
QNCB	BostonLab	9000	9001	9002	9004	

1 of 1 pages (2 items)

129

# Create Node Links

QMS Wep App X +

https://localhost:6443/nodes/kms/56036098-0ba2-4f46-9dc7-f8443c142d21 140% ⭐

Home > Nodes > KMS Nodes > KMS

network-administrator

IDQ FROM VISION TO TECHNOLOGY

KMS

Overview Events Providers Consumers KMS Links

EXPORT CONFIGURATION EDIT KMS NODE DELETE KMS NODE

1) Select KMS Links

KMS Name QNCA

KMS Id 10000001

KMS unique ID cfa8658d9b4744cab5d550c5a7360bb1

Exchange port 9001

Internal port 9000

V.1.0.2.0

# Create Node Links



QMS Wep App X +

https://localhost:6443/nodes/kms/56036098-0ba2-4f46-9dc7-f8443c142d21 140% ⭐

Home > Nodes > KMS Nodes > KMS

English network-administrator

IDQ FROM VISION TO TECHNOLOGY

Dashboard

Nodes

Paths

Monitoring

Firmware

Settings

Users

V.1.0.2.0

KMS

Overview Events Providers Consumers KMS Links

CREATE KMS LINKS

Click Create KMS Links

Name Source Destination Weight Actions

KMS Links

# Create Node Links

KMS link creation step 1/2

Informations

KMS Link name  
QNCAQNCBLINK

Select a destination  
QNCB

1) Name the KMS to KMS communication link

2) Choose the peer Node to the Node you selected

3) Click Next

NEXT

V.1.0.2.0

# Create Node Links

QMS Wep App https://localhost:6443/nodes/kms/56036098-0ba2-4f46-9dc7-f8443c142d21 140% X

## KMS link creation step 2/2

Note:  
The interface you choose depends on the architecture you created. KMS link IP may be a shared “any” IP, MGT IP, KMS IP, or QNC IP

Network interfaces

1) Choose the CerberisXG Alice KMS interface IP or Cerberis3 QNC IP

Source QNCA  
Select a the source IP  
192.168.30.1

2) Choose the CerberisXG Bob KMS interface IP or Cerberis3 QNC IP

Destination QNCB  
Select the destination IP  
192.168.30.2

3) Click Create

PREVIOUS CREATE

V.1.0.2.0

133

# Create Node Links

SWISS  
QUANTUM+

The screenshot shows a web application interface for managing network nodes. On the left, a sidebar lists navigation options: Dashboard, Nodes (highlighted in red), Paths, Monitoring, Firmware, Settings, and Users. The main content area has a breadcrumb trail: Home > Nodes > KMS Nodes > KMS. The title "KMS" is displayed above a navigation bar with tabs: Overview, Events, Providers, Consumers, and KMS Links (the active tab). A green success message box in the top right corner says "✓ KMS Link Item Created". The "KMS Links" section contains a table with columns: Name, Source, Destination, Weight, and Actions. One row is highlighted with a red border, showing "QNCAQNCBLINK" as the name, "QNCA" as the source, "QNCB" as the destination, and "0" as the weight. The "Actions" column for this row includes three icons: a magnifying glass, a pencil, and a downward arrow. Two teal-colored callout boxes with arrows point to these icons: one points to the magnifying glass icon with the text "Verify the link you created", and another points to the downward arrow icon with the text "Edit if needed". The bottom left corner of the page displays the version "V.1.0.2.0".

Name	Source	Destination	Weight	Actions
QNCAQNCBLINK	QNCA	QNCB	0	

# Create Providers



# Create Providers

SWISS  
QUANTUM+

QMS Wep App x +

https://localhost:6443/nodes/providers

Home > Nodes > Providers

?

English

network-administrator

Groups

BostonLab

Nodes

Groups KMS Providers Consumers

CREATE PROVIDER IMPORT PROVIDER

1) Select Nodes

2) Select Providers

3) Click Create Provider

Notice the Group we are configuring

Nodes

Dashboard

Nodes

Paths

Monitoring

Firmware

Settings

Users

EXCEL EXPORT

Name Type KMS Name KMS Id Group name Actions

Items per page

0 of 0 pages (0 items)

V.1.0.2.0

# Create Providers

QMS Wep App

https://localhost:6443/nodes/providers

Provider Creation step 1/4

Note:  
A Provider is the entity providing the keys to the KMS. This is the AUX interface 2 on CerberisXG or QKD module on the Cerberis3

Global information

Select a Type  
Cerberis

Provider name  
QKDA

1) Select “Cerberis3” for both Cerberis3 or Clavis3. Do not select Clavis300.

2) Choose a name for the Provider

3) Click Next

NEXT

V.1.0.2.0

# Create Providers

Provider Creation step 2/4

Select the Provider IP

IP address  
192.168.20.1

1) Enter the IP address of the CerberisXG AUX Interface 2 – or Cerberis3 QKD IP address

2) Click Next

PREVIOUS

NEXT

V.1.0.2.0

# Create Providers

QM Wep App

https://localhost:6443/nodes/providers

140%

## Provider Creation step 3/4

Note: This may be the MGT interface, or KEY interface for CerberisXG depending on prior configuration. This may be the QNC IP or Frontpane IP for Cerberis3.

Related KMS Node

Select the related KMS

QNCA

1) Select the Node that this Provider belongs to

Select the KMS interface

192.168.10.1 (Key)

2) Select the corresponding interface where encryptors are connected to

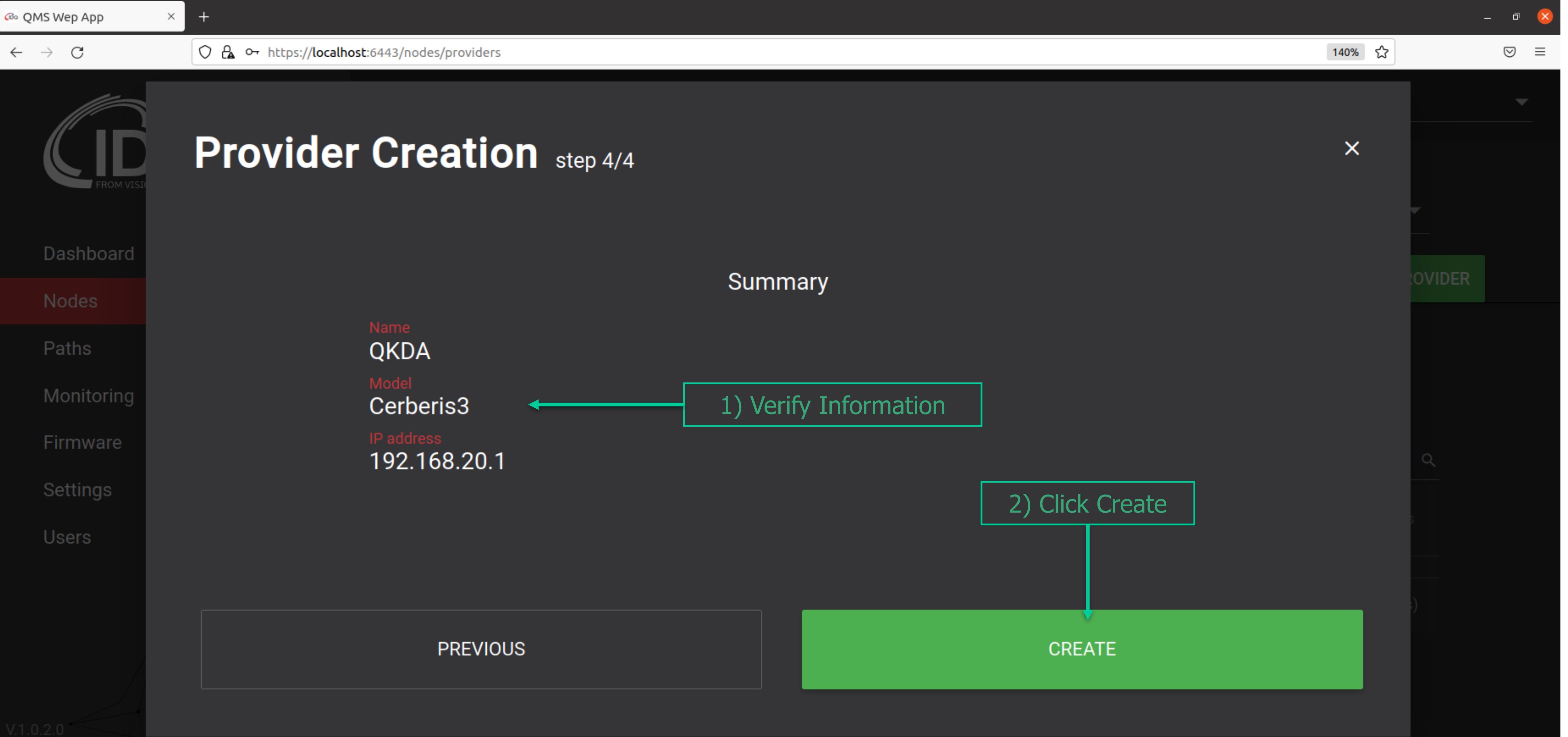
3) Click Next

PREVIOUS

NEXT

V.1.0.2.0

# Create Providers

QMS Wep App https://localhost:6443/nodes/providers 140% 

## Provider Creation step 4/4

Summary

Name	QKDA
Model	Cerberis3
IP address	192.168.20.1

1) Verify Information

2) Click Create

PREVIOUS

CREATE

# Create Providers

QMS Wep App X +

https://localhost:6443/nodes/providers 140% ⭐

Dashboard Nodes PROVIDER

Paths Monitoring Firmware Settings Users

V.1.0.2.0

## Provider Created

You can now choose among the following actions

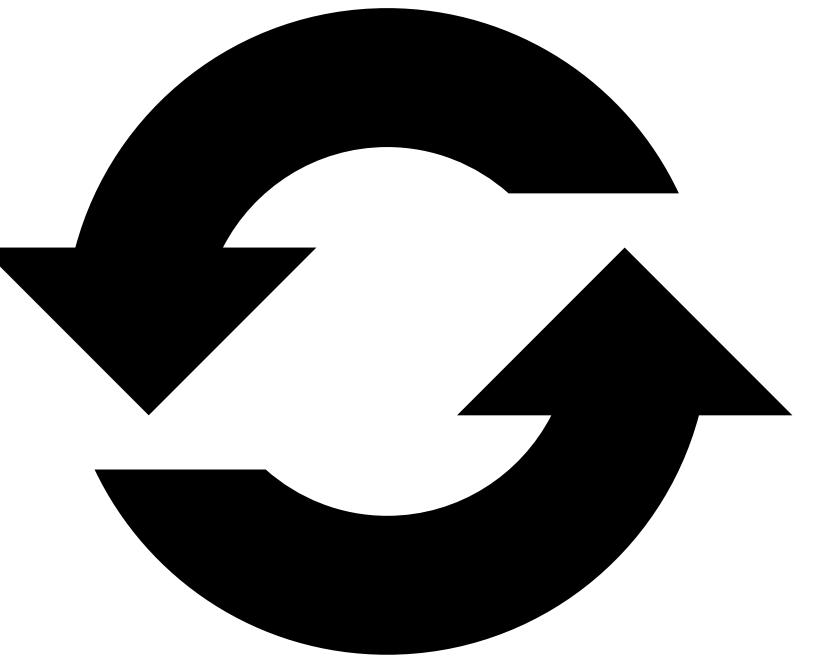
 View provider

 Edit Provider

 Create another provider

Click Create Another Provider

Repeat the same process to create the peer Provider



# Create Providers

Provider Creation step 1/4

Global information

Select a Type: Cerberis

Provider name: QKDB

NEXT

V.1.0.2.0

Provider Creation step 2/4

Select the Provider IP

IP address: 192.168.20.2

PREVIOUS NEXT

V.1.0.2.0

Provider Creation step 3/4

Related KMS Node

Select the related KMS: QNCB

Select the KMS interface: 192.168.10.2 (Key)

PREVIOUS NEXT

V.1.0.2.0

Provider Creation step 4/4

Summary

Name: QKDB

Model: Cerberis3

IP address: 192.168.20.2

PREVIOUS CREATE

V.1.0.2.0

# Create Providers

QMS Wep App

https://localhost:6443/nodes/providers

Provider Created

Click View Provider

You can now choose among the following actions

View provider

Edit Provider

Create another provider

V.1.0.2.0

A screenshot of a web browser window titled "QMS Wep App". The URL in the address bar is "https://localhost:6443/nodes/providers". The main content of the page is a dark-themed interface with a central message: "Provider Created". Below this message, there is a callout box with a cyan border containing the text "Click View Provider". A green arrow points from this text to the first button in a row of three. The first button has an icon of three vertical lines and is labeled "View provider". The second button has an icon of crossed wrenches and is labeled "Edit Provider". The third button has an icon of a star with a trail and is labeled "Create another provider". To the left of the main content area, there is a sidebar with the following menu items: Dashboard, Nodes (which is highlighted in red), Paths, Monitoring, Firmware, Settings, and Users. At the bottom left of the page, the text "V.1.0.2.0" is visible. The bottom right corner features the ID Quantique logo.

# Create Providers

SWISS  
QUANTUM+

QMS Wep App x +

https://localhost:6443/nodes/providers

Home > Nodes > Providers

?

English

network-administrator

Groups

BostonLab

Nodes

Groups KMS Providers Consumers

CREATE PROVIDER IMPORT PROVIDER

Dashboard

Nodes

Paths

Monitoring

Firmware

Settings

Users

EXCEL EXPORT

Search

Actions

Name Type KMS Name KMS Id Group name

QKDA Cerberis3 QNCA 10000001 BostonLab

QKDB Cerberis3 QNCB 10000002 BostonLab

Items per page

1 of 1 pages (2 items)

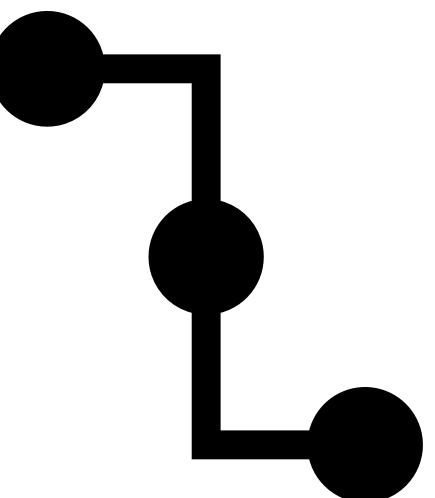
V.1.0.2.0

Verify newly-created Providers

Edit if needed

Name	Type	KMS Name	KMS Id	Group name	Actions
QKDA	Cerberis3	QNCA	10000001	BostonLab	
QKDB	Cerberis3	QNCB	10000002	BostonLab	

# Create Provider Links



# Create Provider Links

QMS Wep App

https://localhost:6443/nodes/kms/56036098-0ba2-4f46-9dc7-f8443c142d21

Home > Nodes > KMS Nodes > KMS

English

network-administrator

KMS

Overview Events Providers Consumers KMS Links

CREATE PROVIDER IMPORT PROVIDER

Dashboard

Nodes

Paths

Monitoring

Firmware

Settings

Users

V.1.0.2.0

1) Select Nodes

2) Select Providers

3) View Provider

Name	Type	KMS Name	KMS Id	Actions
QKDA	Cerberis3	QNCA	10000001	

Items per page: 10

1 of 1 pages (1 items)

# Create Provider Links



QMS Wep App

https://localhost:6443/nodes/providers/20000001

Home > Nodes > Providers > Provider

English

network-administrator

Dashboard

Nodes

Paths

Monitoring

Firmware

Settings

Users

V.1.0.2.0

**Provider**

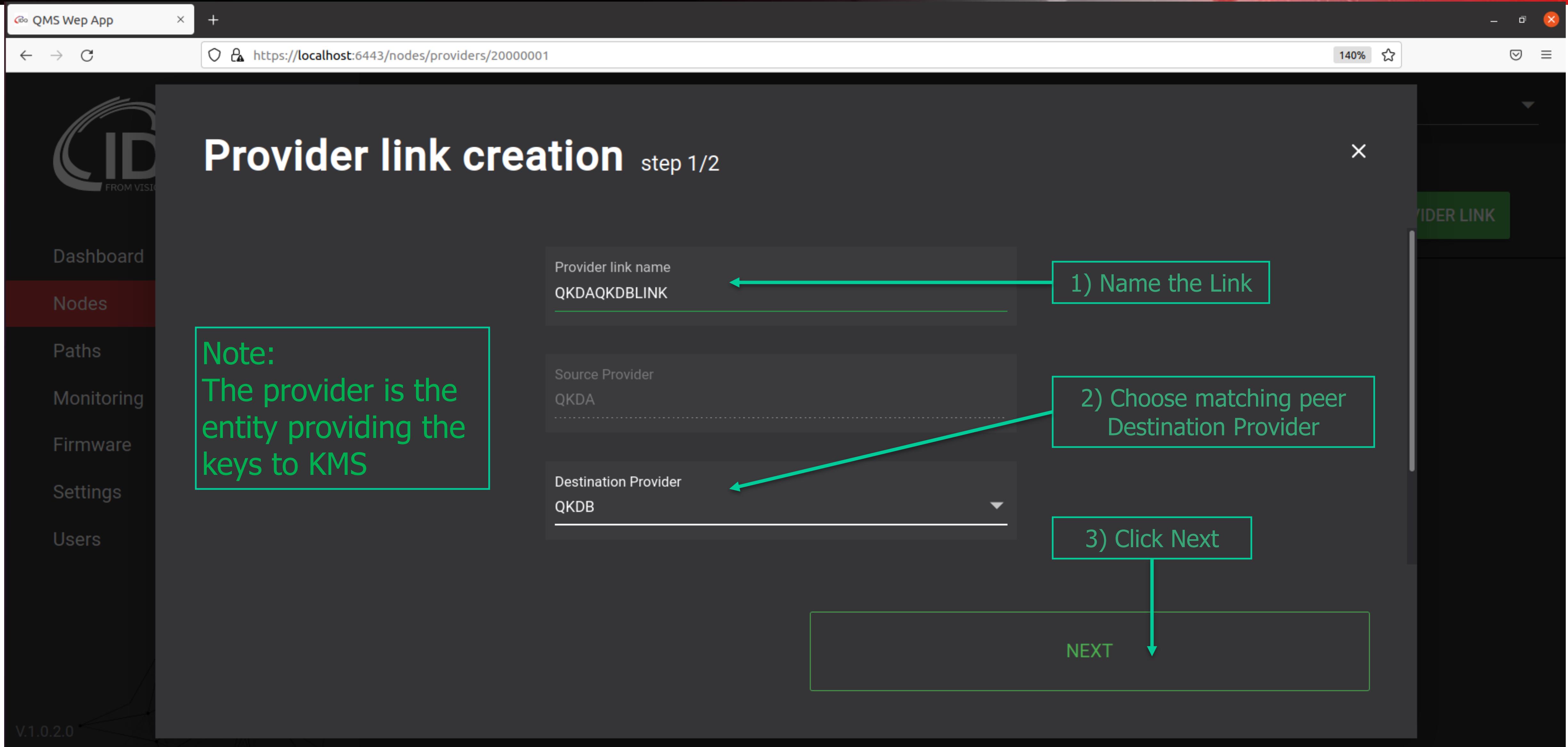
Overview Provider link Events

**CREATE PROVIDER LINK**

**Click Create Provider Link**

A screenshot of a web browser displaying the ID Quantique QMS Web Application. The URL in the address bar is https://localhost:6443/nodes/providers/20000001. The page title is "Provider". The navigation menu on the left includes "Dashboard", "Nodes" (which is highlighted in red), "Paths", "Monitoring", "Firmware", "Settings", and "Users". The version number "V.1.0.2.0" is visible at the bottom left. The main content area shows tabs for "Overview", "Provider link" (which is active and highlighted in red), and "Events". A large green button labeled "CREATE PROVIDER LINK" is located on the right. A cyan callout box with the text "Click Create Provider Link" has an arrow pointing towards the green button. The top right corner of the browser window shows a zoom level of 140%.

# Create Provider Links



QMS Wep App https://localhost:6443/nodes/providers/20000001 140% ID

## Provider link creation step 1/2

Provider link name  
QKDAQKDBLINK

Source Provider  
QKDA

Destination Provider  
QKDB

Note:  
The provider is the entity providing the keys to KMS

1) Name the Link

2) Choose matching peer Destination Provider

3) Click Next

NEXT

# Create Provider Links

SWISS  
QUANTUM+

QMS Wep App https://localhost:6443/nodes/providers/20000001 140%  PROVIDER LINK

## Provider link creation step 2/2

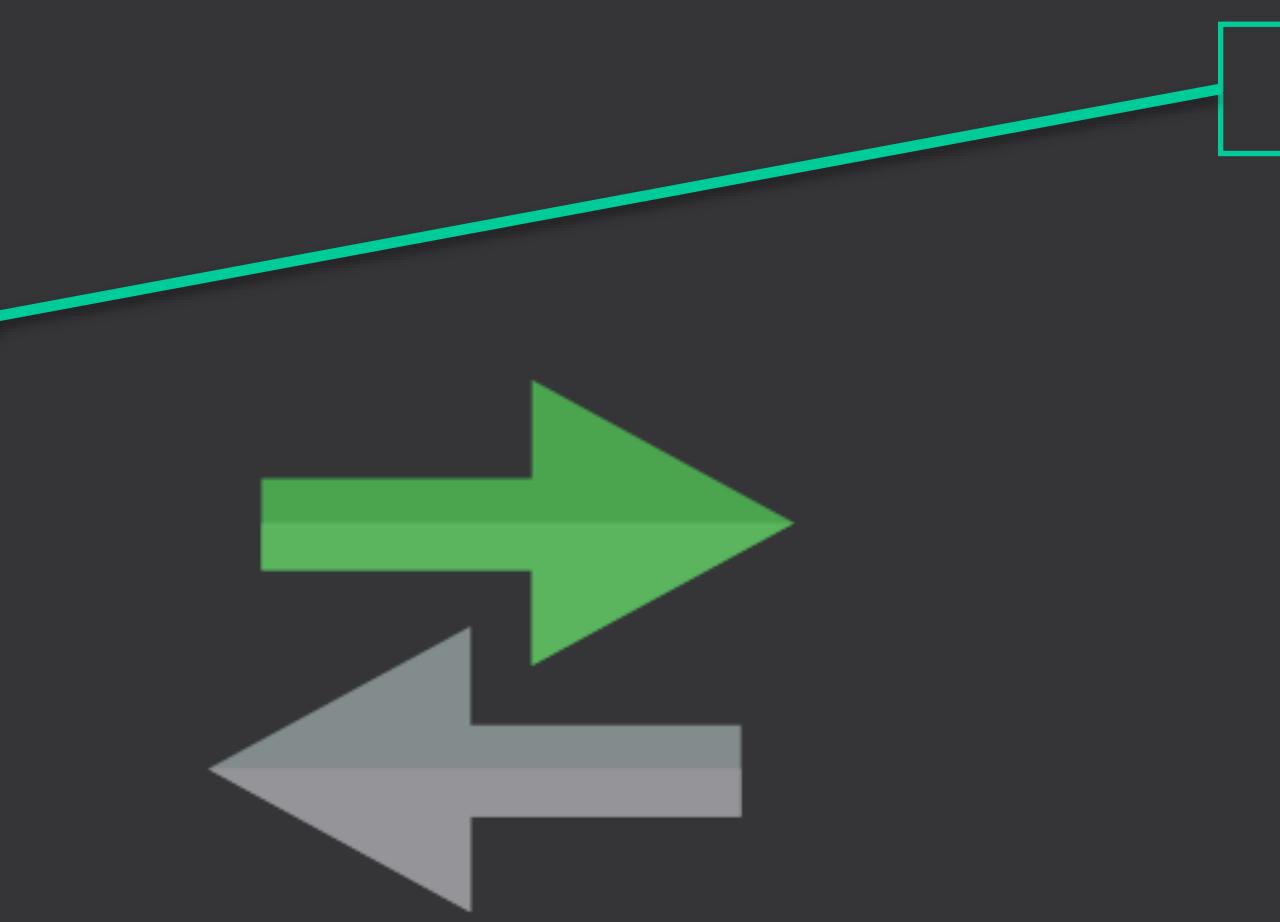
1) Verify Information

Name	QKDA	Name	QKDB
Related Kms Name	QNCA	Related Kms Name	QNCB
Model	Cerberis3	Model	Cerberis3
Type	Alice	Type	Bob

2) Click Create

PREVIOUS CREATE

V.1.0.2.0



# Create Provider Links

QMS Wep App X +

https://localhost:6443/nodes/providers/20000001 140% ☆

Home > Nodes > Providers > Provider

English network-administrator

IDQ FROM VISION TO TECHNOLOGY

Provider Overview Provider link Events

**EDIT PROVIDER LINK** **DELETE PROVIDER LINK**

Dashboard

Nodes Paths Monitoring Firmware Settings Users

Name QKDA Related Kms Name QNCA Type Alice

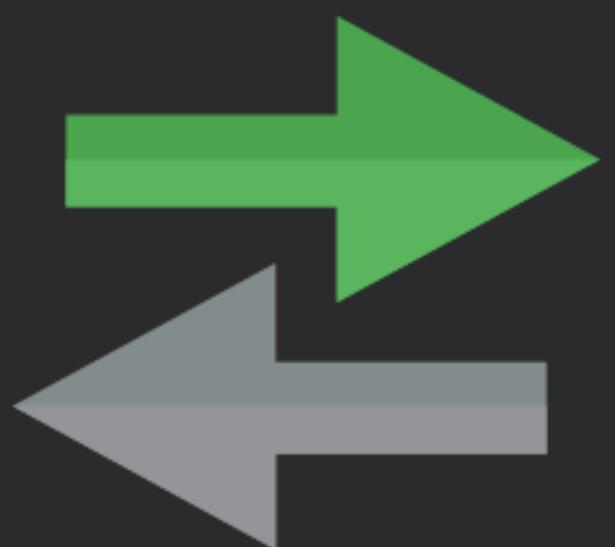
Name QKDB Related Kms Name QNCB Type Bob

Link Name QKDAQKDBLINK

Key configurations Value

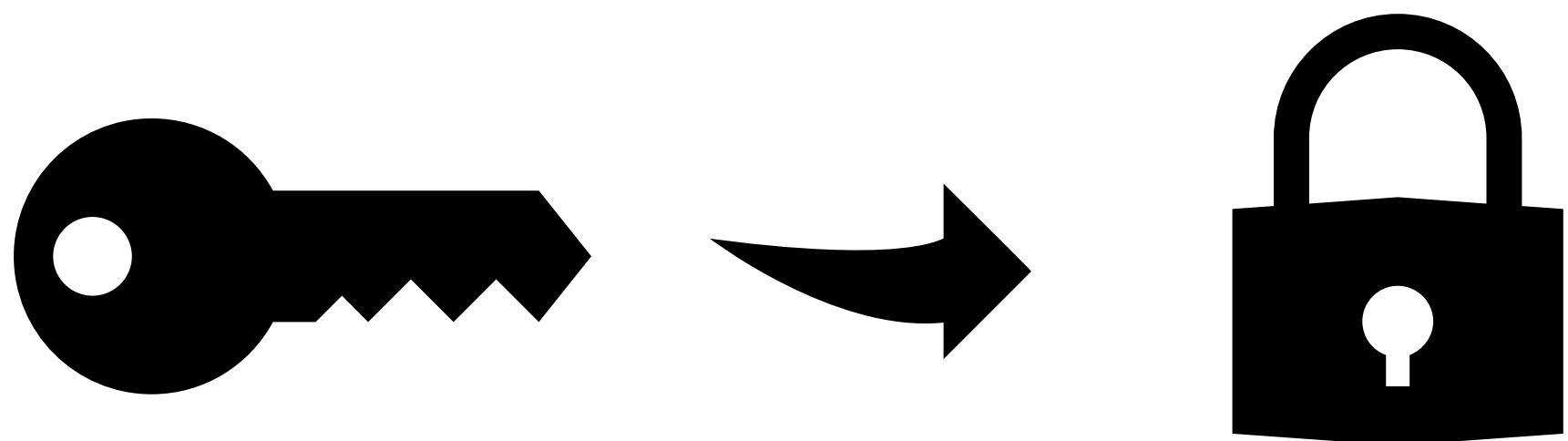
Max key count 100

Verify newly-created provider link



V.1.0.2.0

# Create Consumer



# Create Consumer

SWISS  
QUANTUM+

The screenshot shows the QMS Web App interface with the following details:

- Header:** QMS Wep App, https://localhost:6443/nodes/consumers, 140%, network-administrator.
- Sidebar:** IDQ logo, FROM VISION TO TECHNOLOGY, Dashboard, Nodes (highlighted in red), Paths, Monitoring, Firmware, Settings, Users, V.1.0.2.0.
- Breadcrumbs:** Home > Nodes > Consumers.
- Top Navigation:** Groups, KMS, Providers, Consumers (highlighted).
- Right Sidebar:** Groups (BostonLab), CREATE CONSUMER, IMPORT CONSUMER.
- Content Area:** Consumers, EXCEL EXPORT, Name, Type, KMS Name, KMS Id, Group name, Actions, Items per page (10), 0 of 0 pages (0 items).

Three green boxes with arrows indicate the steps to create a consumer:

- 1) Select Nodes (points to the Nodes sidebar item)
- 2) Select Consumers (points to the Consumers tab in the top navigation)
- 3) Click Create Consumer (points to the green button labeled "CREATE CONSUMER")

# Create Consumer

QMS Wep App https://localhost:6443/nodes/consumers 140% 140%

## Consumer Creation step 1/3

Global informations

1) Select encryptor type - key delivery protocol

2) Configure a name for the encryptor. This name also becomes the address/ID

3) Click Next

Dashboard

Nodes

Paths

Monitoring

Firmware

Settings

Users

V.1.0.2.0

SELECT A TYPE

ETSI

Consumer name

ETSI-A

NEXT

The screenshot shows a web application interface for creating a consumer. On the left, there's a sidebar with links: Dashboard, Nodes (which is highlighted in red), Paths, Monitoring, Firmware, Settings, and Users. The main area has a dark background with white text. At the top, it says 'Consumer Creation step 1/3'. Below that, it says 'Global informations'. There are three numbered callouts in green boxes with arrows pointing to specific fields:

- 1) Select encryptor type - key delivery protocol: Points to a dropdown menu labeled 'Select a Type' with 'ETSI' selected.
- 2) Configure a name for the encryptor. This name also becomes the address/ID: Points to a text input field labeled 'Consumer name' containing 'ETSI-A'.
- 3) Click Next: Points to a large green button labeled 'NEXT'.

The bottom right corner of the screen features the ID Quantique logo."/>

# Create Consumer

SWISS  
QUANTUM+

QMS Wep App

https://localhost:6443/nodes/consumers

140% 140%

Consumer Creation step 2/3

KMS node

Related KMS Node

KMS Ip Address

1) Select the Node that this encryptor is attached to

2) Select the interface IP where this node is requesting keys from

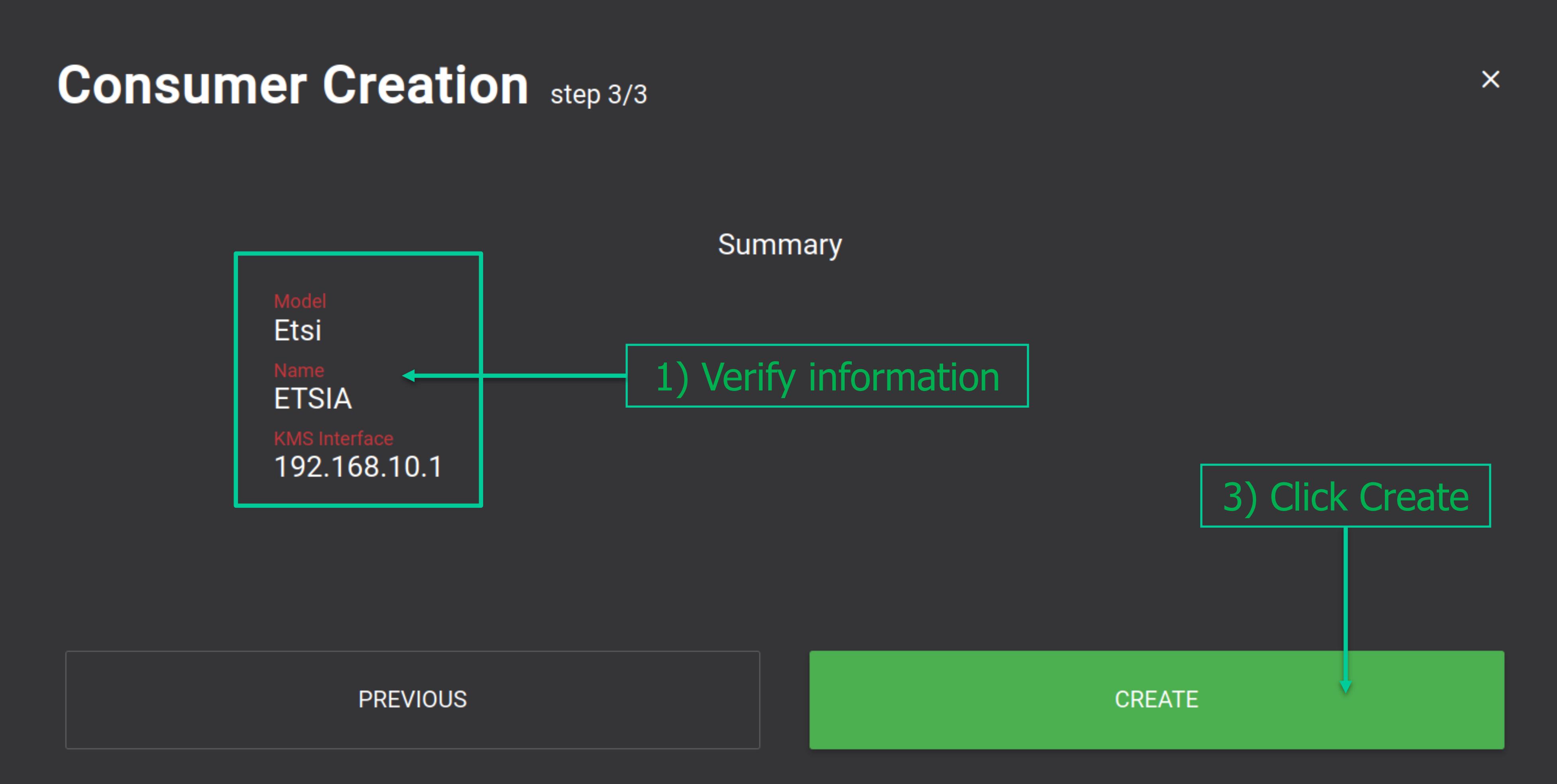
3) Click Next

PREVIOUS

NEXT

V.1.0.2.0

# Create Consumer

QMS Wep App https://localhost:6443/nodes/consumers 140% 

## Consumer Creation step 3/3

Summary

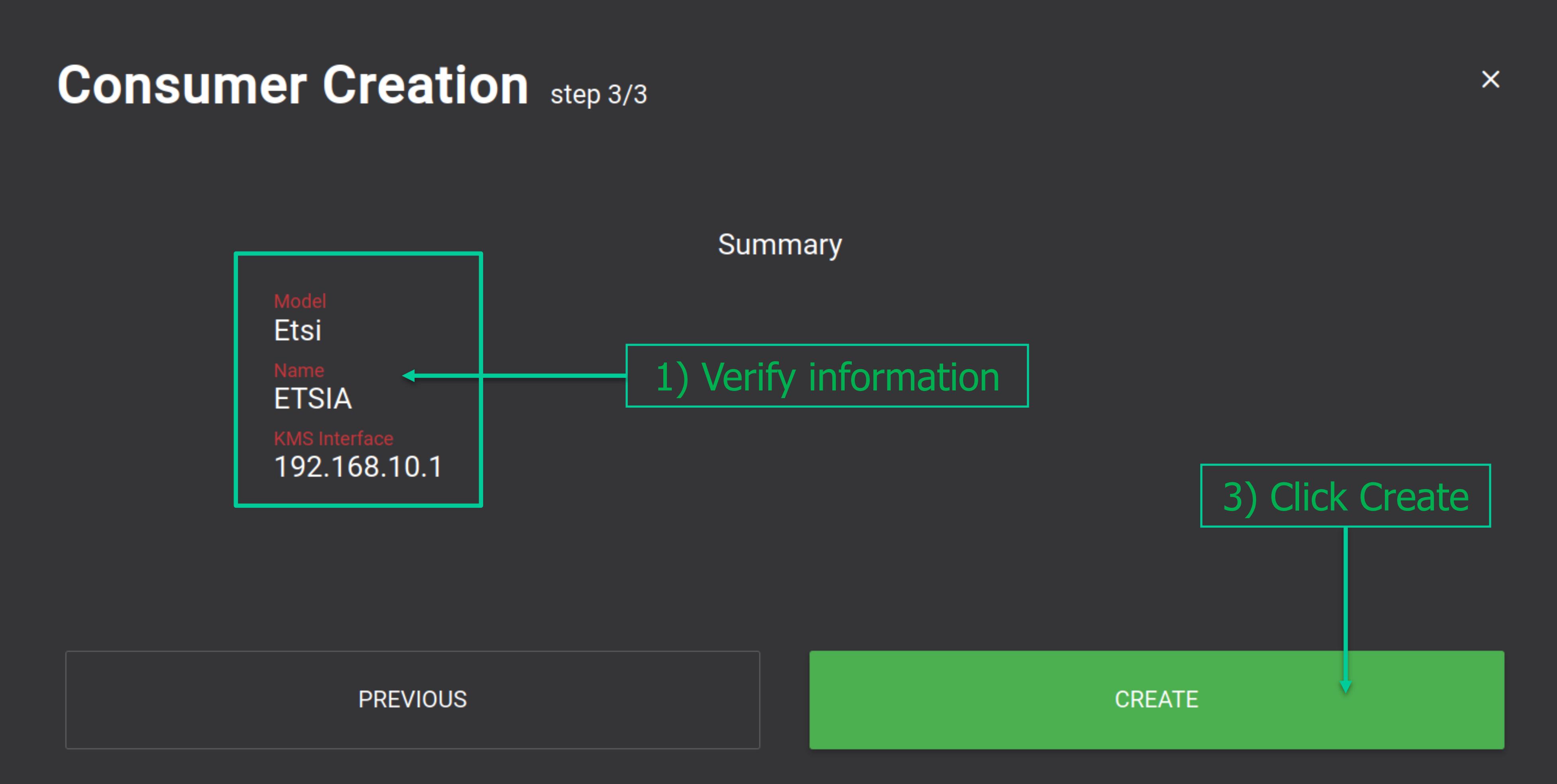
Model: Etsi  
Name: ETSIA  
KMS Interface: 192.168.10.1

1) Verify information

3) Click Create

PREVIOUS CREATE

V.1.0.2.0



# Create Consumers

SWISS  
QUANTUM+

QMS Wep App X +

← → C https://localhost:6443/nodes/consumers 140% ☆ ☰ ☱ ☲

Dashboard

Nodes

Paths

Monitoring

Firmware

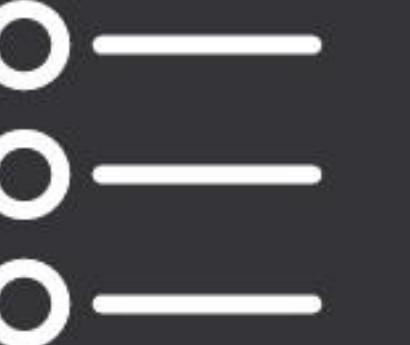
Settings

Users

V.1.0.2.0

# Consumer Created

You can now choose among the following actions



Consumers



Edit Consumer



Create another consumer

Click Edit Consumer. We need to add the certificate information

# Create Consumers

QMS Wep App x +

https://localhost:6443/nodes/consumers/30000001/edit 140% ☆

Home > Nodes > Consumers > Consumer > Edit ? English network-administrator

Edit consumer's configuration

EXPORT CONFIGURATION IMPORT CONFIGURATION CANCEL SAVE

Agent information Inside Kms

SubjectDn  
c=CH,ST=Geneva State,L=Geneva,O=ID Quantique,OU=QuantumSa

Process Name  
CAGTR

Key Type  
QKEY

Process port

1) Scroll down and configure Subject DN of the client/consumer certificate. See next slide for more information.

2) Click Save

# Create Consumers

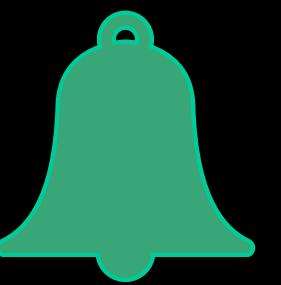
The SubjectDN is not needed for Cisco Encryptors.

To view the SubjectDN, open the certificate with openssl or other certificate management tool:

Example:

```
openssl x509 -in Client1.pem -text -noout
```

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



Please see next slide for additional  
important information

# Create Consumers



SubjectDn

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

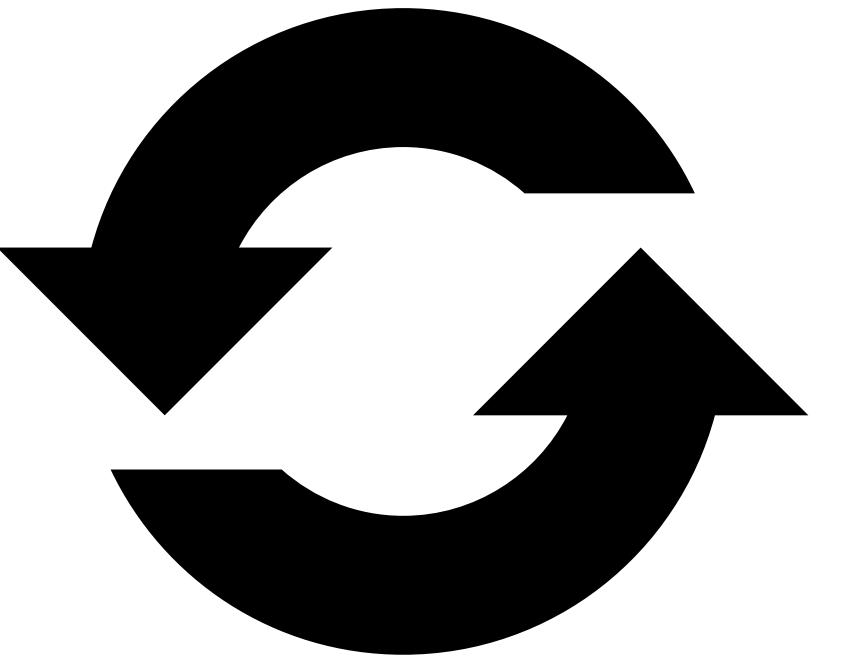
Example: openSSL ouput:

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

You need to import **the following Subject DN into QMS**

C=CH,ST=Geneva State,L=Geneva,O=ID Quantique,OU=QuantumSafe,CN=enc-bob

Repeat the same process to create the peer  
Consumer



# Create Consumers

SWISS  
QUANTUM+

The screenshot displays the ID Quantique QMS Web Application interface, illustrating the consumer creation process through four distinct steps:

- Step 1/3: Consumer Creation** (Top Right): A modal window titled "Consumer Creation step 1/3" shows the "Global informations" section. It includes fields for "Name" (set to "Etsi") and "Type" (set to "QKEY"). A "NEXT" button is visible at the bottom right.
- Step 2/3: Consumer Creation** (Bottom Left): A modal window titled "Consumer Creation step 2/3" shows the "KMS node" field set to "QNCB" and the "KMS Ip Address" field set to "192.168.10.2 (Key)". A "NEXT" button is visible at the bottom right.
- Step 3: Edit consumer's configuration** (Center): A modal window titled "Edit consumer's configuration" shows the "Agent information" tab selected. It displays fields for "SubjectDn" (set to "CN=State,L=Geneva,O=ID Quantique,OU=QuantumSafe,CN=enc-bolt"), "Process Name" (set to "CAGTR"), and "Key Type" (set to "QKEY"). Buttons for "EXPORT CONFIGURATION", "IMPORT CONFIGURATION", "CANCEL", and "SAVE" are present.
- Step 4: Nodes Overview** (Top Left): The main application window shows the "Nodes" tab selected. It displays a list of consumers, including one named "Etsi". A green "CREATE CONSUMER" button is located at the top right of this section.

# Create Consumers

QMS Wep App X +

https://localhost:6443/nodes/consumers 140% ⭐

Dashboard

Nodes

Paths

Monitoring

Firmware

Settings

Users

V.1.0.2.0

## Consumer Created

Click Consumers

You can now choose among the following actions

Consumers

Edit Consumer

Create another consumer

A green arrow points from the text "Click Consumers" to the "Consumers" icon.

# Create Consumers

SWISS  
QUANTUM+

QMS Wep App https://localhost:6443/nodes/consumers 140% network-administrator

Home > Nodes > Consumers

Groups English network-administrator

BostonLab

Nodes Groups KMS Providers Consumers

CREATE CONSUMER IMPORT CONSUMER

Verify newly-created consumer information

Consumers EXCEL EXPORT

Edit if needed. Don't forget the SubjectDN information!

Name	Type	KMS Name	KMS Id	Group name	Actions
ETSIA	Etsi	QNCA	10000001	BostonLab	
ETSIB	Etsi	QNCB	10000002	BostonLab	

1 of 1 pages (2 items)

V.1.0.2.0

# Create Path



# Create Consumer Links and Key Routing Path

SWISS  
QUANTUM+

QMS Wep App

https://localhost:6443/paths/100

Home > Paths

English network-administrator

Group: BostonLab

Concerned KMS

DEPLOY GROUP

Dashboard

Nodes

Paths

Monitoring

Firmware

Settings

Users

V.1.0.2.0

Consumer connections

ADD CONNECTION

1) Select Paths

Select Add Connection

# Create Consumer Links and Key Routing Path

SWISS  
QUANTUM+

The screenshot shows the QMS Web App interface for creating consumer links. The main window title is "Consumer link creation step 1/3". The left sidebar has "Paths" selected. The main area is titled "Source selection" and contains two dropdown menus: one for "KMS/QNCA" with "QNCA" selected, and another for "Consumer" with "ETSI" selected. A large green box highlights the first two steps:

- 1) Select Node of encryptor connection
- 2) Select Encryptor

A green arrow points from the text "1) Select Node of encryptor connection" to the "KMS/QNCA" dropdown. Another green arrow points from the text "2) Select Encryptor" to the "Consumer" dropdown. A third green box highlights the "NEXT" button at the bottom right of the main panel.

3) Click Next

NEXT

# Create Consumer Links and Key Routing Path

SWISS  
QUANTUM+

QMS Wep App

https://localhost:6443/paths/100

Getting Started

Home

Paths

Consumer link creation step 2/3

Destination selection

KMS  
QNCB

Consumer  
ETSIB

PREVIOUS

NEXT

1) Select Node of peer encryptor

2) Select peer encryptor

3) Click Next

network-administrator

DEPLOY GROUP

V.1.0.2.0

# Create Consumer Links and Key Routing Path

QMS Wep App × +

Getting Started

https://localhost:6443/paths/100

Home Paths Consumers

Dashboard Nodes Paths Monitoring Firmware Settings Users

Path

Consumer link creation step 3/3

Link name

Name  
ETSLINK

1) Create a name for this link

2) Click Create

PREVIOUS CREATE

V.1.0.2.0

# Create Consumer Links and Key Routing Path

SWISS  
QUANTUM+

The screenshot shows the QMS Web App interface for managing paths. On the left, a sidebar menu includes Options, Dashboard, Nodes, Paths (which is selected and highlighted in red), Monitoring, Firmware, Settings, and Users. The main content area displays 'Consumer connections' with a list item 'ETSLINK' and a sub-item 'QNCA → QNCB'. A green arrow points from the text '1) Verify newly-created Link' to the 'QNCA → QNCB' item. Another green arrow points from the text '2) Click Deploy Group.' to a green button labeled 'DEPLOY GROUP' in the top right corner. A note at the bottom states: 'Note: For V3.1.0, you must set the consumer default key size to 256'.

1) Verify newly-created Link

2) Click Deploy Group.  
This will push the entire configuration for this group to the Nodes.

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

# Verify Topology

SWISS  
QUANTUM+

QMS Wep App

https://localhost:6443/nodes/kms/70623689-73a4-4826-b75e-17016055809e

Home > Nodes > KMS Nodes > KMS

English network-administrator

KMS

Overview Events Providers Consumers KMS Links

EXPORT CONFIGURATION EDIT KMS NODE DELETE KMS NODE

CiscoA CiscoB ETSIA ETSIB CerberisXGA CerberisXGB QKDA QKDB

Visual Topology

```
graph TD; CerberisXGA --- CiscoA; CerberisXGA --- CiscoB; CerberisXGA --- ETSIA; CerberisXGA --- ETSIB; CerberisXGA --- CerberisXGB; CerberisXGA --- QKDA; CerberisXGA --- QKDB; CiscoA --- CiscoB; ETSIA --- ETSIB; CerberisXGB --- QKDA; CerberisXGB --- QKDB;
```

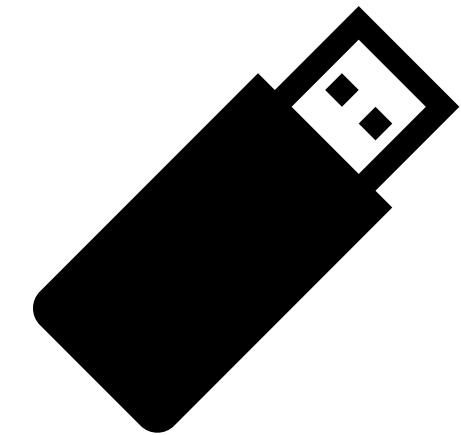
# Certificate Configuration – Root CA



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

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

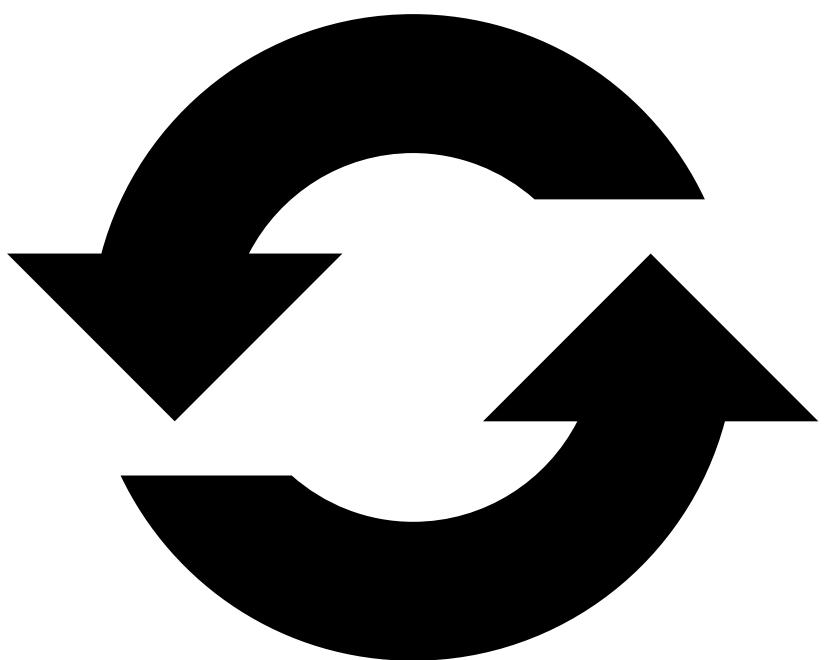
```
admin> encryptor add-client-ca
PEM files:
1) ChrisCA.pem
2) QKDServer.pem
3) QKDServer_pkey.pem
Please select a file (between 1 and 3).
```



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

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

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



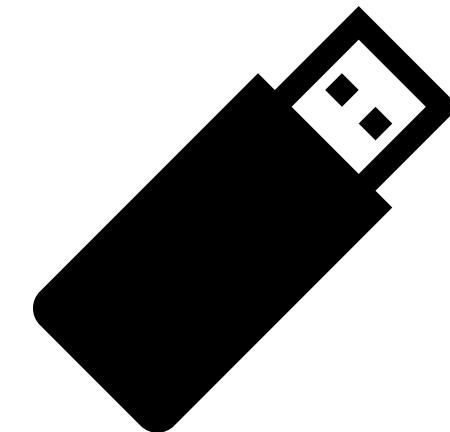
# Certificate Configuration – Root CA



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

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

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



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

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

# Certificate Configuration – Server Certificates/Keys

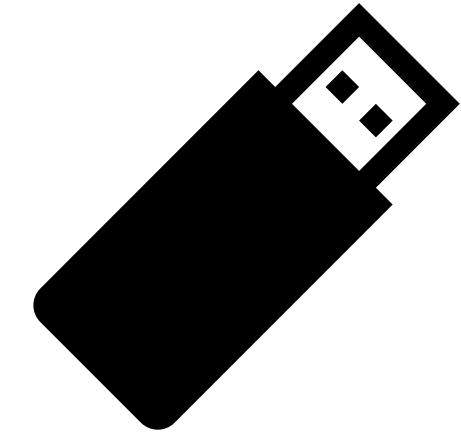
- Load the server Certificate and keys on CerberisXG/QNC:
  - Plug the USB key with the certificates and keys into the CerberisXG/QNC USB port
  - Execute the command in CerberisXG/QNC CLI: **encryptor load-key** then copy server keys and server certificates on the system

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

Usage: encryptor load-key [options]

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

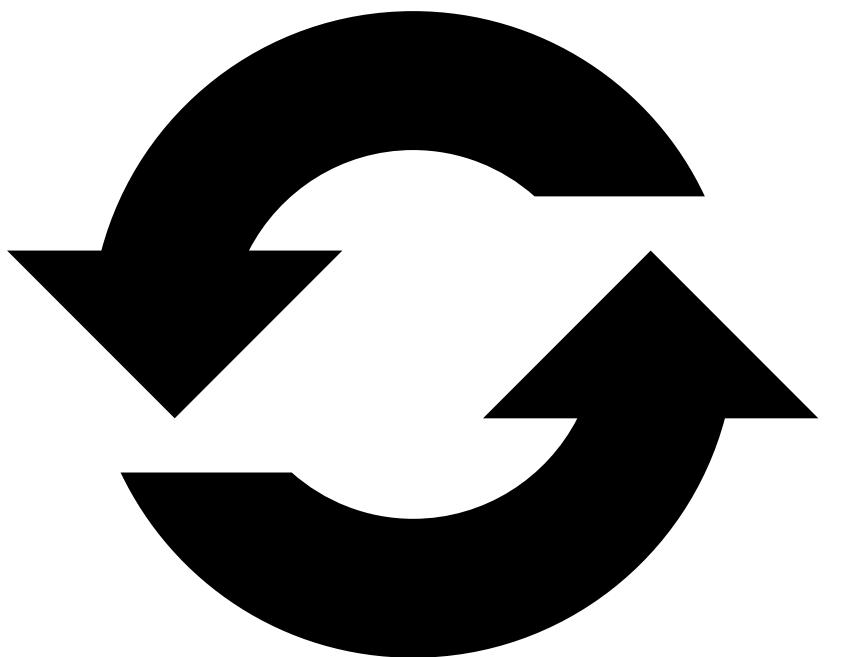
admin> encryptor load-key -o
PEM files:
 1) ChrisCA.pem
 2) QKDServer.pem
 3) QKDServer_pkey.pem
Please select a file (between 1 and 3). 2
Key loaded
admin> encryptor load-key -o
PEM files:
 1) ChrisCA.pem
 2) QKDServer.pem
 3) QKDServer_pkey.pem
Please select a file (between 1 and 3). 3
Key loaded
admin>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



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

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

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



# Certificate Configuration – Server Certificates/Keys

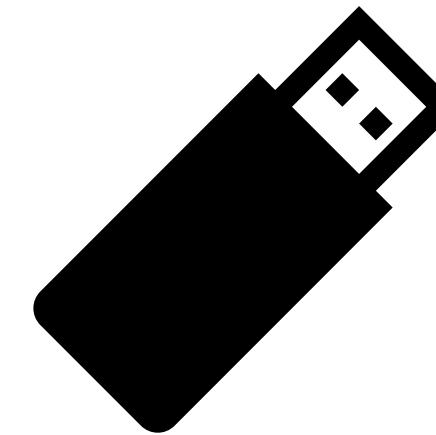
- Load the server Certificate and keys on CerberisXG/QNC:
  - Plug the USB key with the certificates and keys into the CerberisXG/QNC USB port
  - Execute the command in CerberisXG/QNC CLI: **encryptor load-key** then copy server keys and server certificates on the system

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

Usage: encryptor load-key [options]

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

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



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

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

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

# Certificate Configuration – Assign Server Keys



Set the server certificates and keys:

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

**Note: File names must end in .pem extension**

```
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]
```

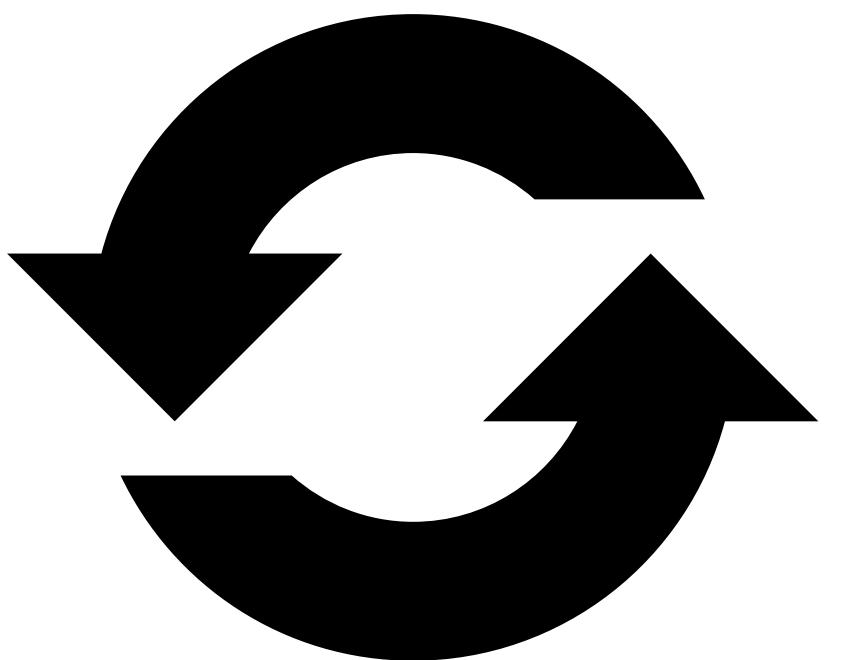
The consumers you created in QMS are displayed

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

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

Use standalone command “encryptor” to verify configuration

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



# Certificate Configuration – Assign Server Keys



Set the server certificates and keys:

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

**Note: File names must end in .pem extension**

```
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]
```

The consumers you created in QMS are displayed

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

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

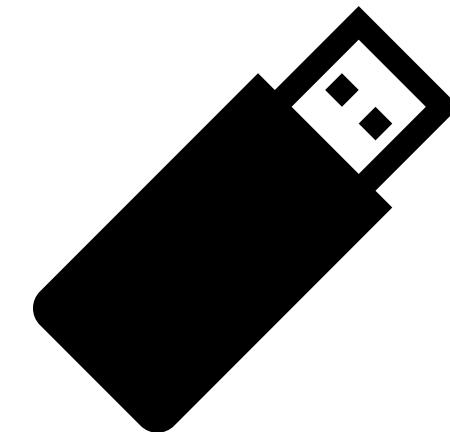
Use standalone command “encryptor” to verify configuration

# Add Initial Authentication Key

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

*authkey generate*

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



*authkey Load*

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

# Verify ISK has been set

View the QKD logs:

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

The system will not start generating quantum keys without it.

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

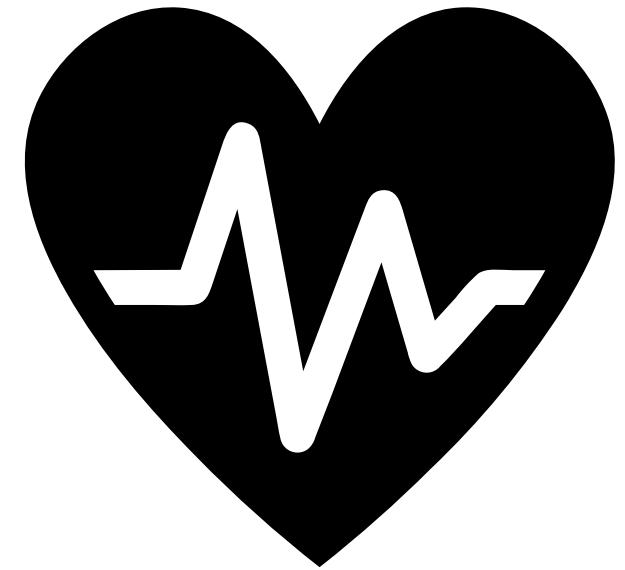
```
2020-02-19T16:36:55.374 DEBUG [(anonymous namespace)::FsmFrontend::NormalOperation_::ExecutingGeneralInitialization_*] Leaving ExecutingGeneralInitialization state.  
2020-02-19T16:36:55.375 DEBUG [(anonymous namespace)::FsmFrontend::NormalOperation_::ExecutingSecurityInitialization*] Entering ExecutingSecurityInitialization 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.  
2020-02-19T16:42:20.768 DEBUG [cervino::communicator::backend::KeyExchangeBob] Starting to...
```

# Key Generation

```
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
5, DATA delay = 2409 ps, MONITOR delay = 2200 ps.
```

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

## Monitor vital system statistics



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

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

Verify clid and idq4p options enabled

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

See QNET Tool User Guide for complete list of monitoring options

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

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

Once for each QKD system

```
christian@christian-VirtualBox:~/Documents/qnet$ qnet monitor node CerberisXGA -clid -idq4p
KMS node CerberisXGA monitoring configuration successfully updated.
christian@christian-VirtualBox:~/Documents/qnet$ qnet monitor node CerberisXGB -clid -idq4p
KMS node CerberisXGB monitoring configuration successfully updated.
christian@christian-VirtualBox:~/Documents/qnet$
```

# Monitoring/Statistics Preparation

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

## 1. Via SSH connection:

Log-in as admin in the QNET Shell of each node:

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

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

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

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

## 2. With QNET Tools:

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

# Monitoring/Statistics

The screenshot shows the QMS Web App interface for monitoring. On the left, a sidebar menu includes Dashboard, Nodes, Paths, Monitoring (which is selected and highlighted in red), Firmware, Settings, and Users. A large green box labeled "Choose Monitoring" has an arrow pointing to the "Monitoring" button in the sidebar. At the top, the navigation bar shows Home > Monitoring > Events. The main content area is titled "Monitoring" and displays a table of event logs. The table columns are Occured Time, Node Name, Subsystem, Category, Type, and Message. The first few rows show events like "Process Stop(system QNCB process EBRK STOP)" and "Link Disconnect(QNCB link QNCB(192.168.10.106) <-> QNCA(192.168.10.101))". A green box labeled "Select Events, Statistics, or Alarms" has an arrow pointing to the "Alarms" tab in the top navigation. Another green box labeled "Use Filters as Needed" has an arrow pointing to the filter icons in the table header. The bottom of the page shows pagination and a total of 17 pages with 162 items.

Occured Time	Node Name	Subsystem	Category	Type	Message
7/23/2021 1:02:37 PM	QNCB	EBRK	STOP	WARNING	Process Stop(system QNCB process EBRK STOP)
7/23/2021 1:02:36 PM	QNCB	EBRK	DISCONNECT	WARNING	Link Disconnect(QNCB link QNCB(192.168.10.106) <-> QNCA(192.168.10.101))
7/23/2021 9:58:29 AM	QNCA	SMMD	2	WARNING	310
7/23/2021 9:58:30 AM	QNCB	SMMD	2	WARNING	310
7/23/2021 9:58:19 AM	QNCA	SMMD	3	WARNING	310
7/23/2021 9:58:20 AM	QNCB	SMMD	3	WARNING	310
7/23/2021 9:57:00 AM	QNCA	QKRD	2	WARNING	311
7/23/2021 9:57:00 AM	QNCB	QKRD	2	WARNING	311
7/23/2021 9:56:53 AM	QNCA	QKRD	2	WARNING	311
7/23/2021 9:56:54 AM	QNCB	QKRD	2	WARNING	311

# Generating Custom Graphs/Statistics

The screenshot shows the QMS Wep App interface for monitoring statistics. A vertical teal arrow on the left points from the 'Monitoring' link in the sidebar to the 'Statistics' tab in the top navigation bar. A horizontal teal arrow points from the 'Statistics' tab to the 'Nodes' series selection. A second horizontal teal arrow points from the 'Nodes' selection to the graph area.

**1) Select Monitoring**

**2) Select Statistics**

**3) Choose Series**

QMS Wep App - https://localhost:6443/monitoring/StatisticsView

Home > Monitoring > Statistics

Monitoring Events Statistics Alarms

Dashboard Nodes Paths Monitoring Firmware Settings Users

Graph configuration  
Choose a start Date 7/26/2021 11:51 AM  
Choose an end Date 7/26/2021 1:51 PM FILTER

Auto Refresh Last Updated: 1:51:06 PM

Nodes

+ ADD A GRAPH

QKDFFPGA\_QBER on qkda,qkdb

QKDFFPGA\_QBER

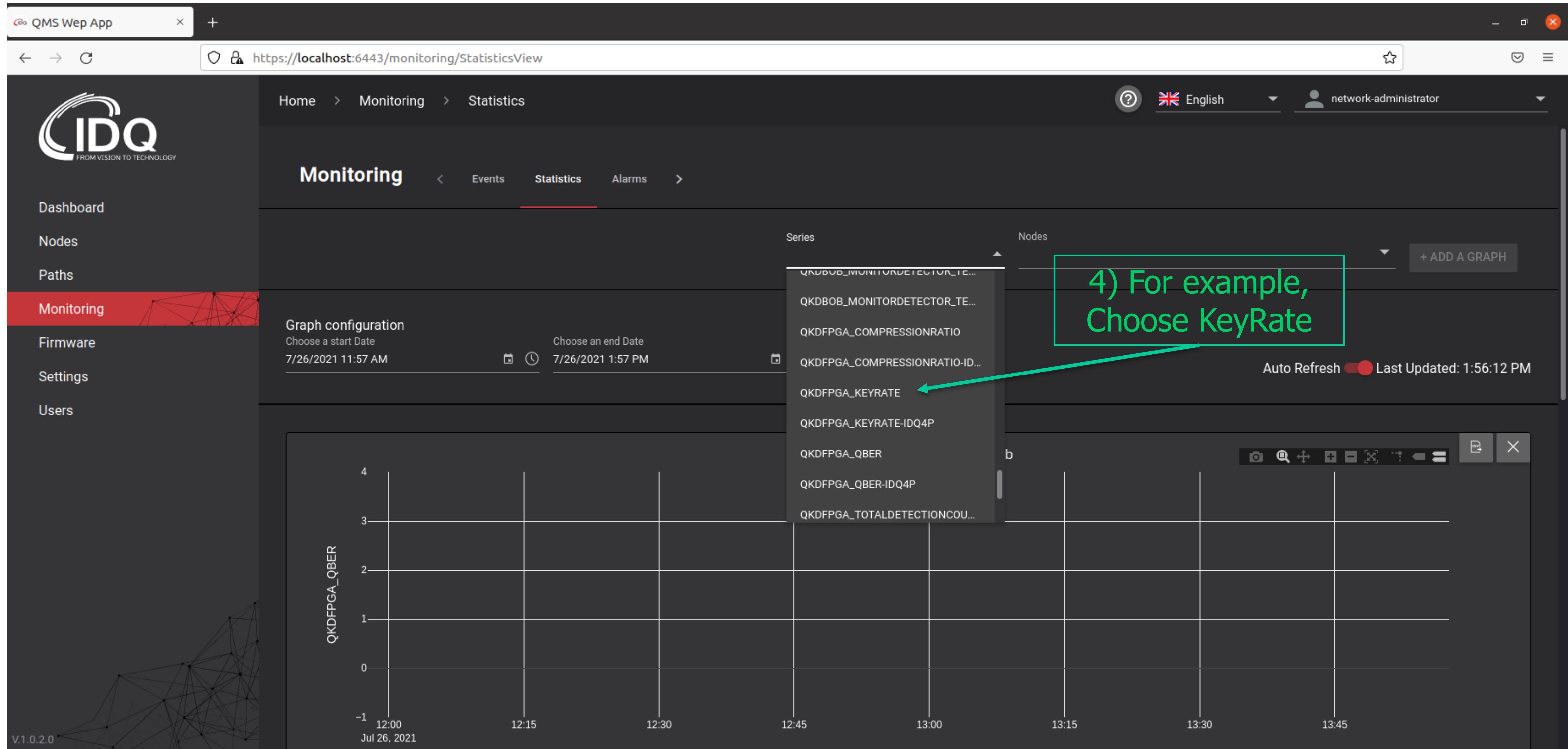
V.1.0.2.0

12:00 12:15 12:30 12:45 13:00 13:15 13:30 13:45

Jul 26, 2021

189

# Generating Custom Graphs/Statistics



# Generating Custom Graphs/Statistics

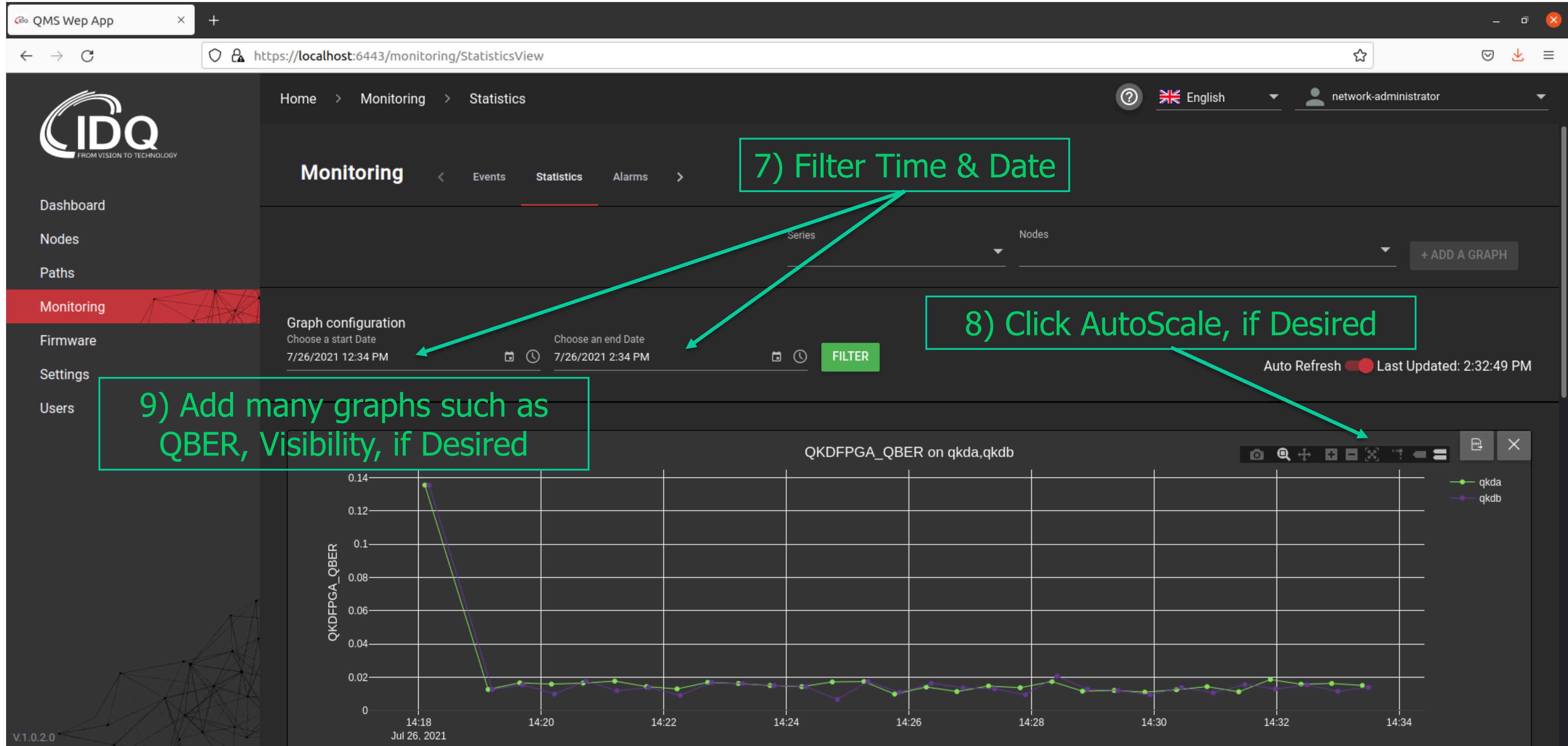
The screenshot shows the ID Quantique QMS Web App interface. The left sidebar has a red-highlighted 'Monitoring' section. The main area is titled 'Monitoring' and shows a graph configuration window. The graph title is 'QKD FPGA\_QBER on qkda,qkdb'. The Y-axis is labeled 'QKD FPGA\_QBER' with values from 0 to 4. The X-axis shows time from 12:00 to 13:45 on Jul 26, 2021. A single data series 'QKD FPGA\_KEYRATE' is selected. To the right, a list of nodes is shown: centaurisa, centaurisb, etsia, etsib, qkda, qkdb, qnca, qncc. A green box labeled '5) Choose (and add) Correct Modules' highlights the node list. Another green box labeled '6) Click on "Add Graph"' highlights the '+ ADD A GRAPH' button.

5) Choose (and add) Correct Modules

+ ADD A GRAPH

6) Click on "Add Graph"

# Generating Custom Graphs/Statistics



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



# Add a Template



# Email Notifications – Event & Alarm Triggers

The screenshot shows the QMS Web App interface with a dark theme. On the left is a sidebar with links: Dashboard, Nodes, Paths, Monitoring, Firmware, Settings (which is highlighted in red), and Users. The main content area has a breadcrumb navigation: Home > Settings > Notification configuration. The top navigation bar includes a logo for IDQ, language selection (English), and user information (network-administrator). The main content is titled "Settings" and shows three tabs: Notification configuration (selected), Maps, and TimeSeries. Below these tabs are three sub-links: Notification configurations, Distribution list, and Template. The "Template" section is currently active, showing a table with one row: "No records to display". At the bottom of this section is a green button labeled "ADD TEMPLATE". Three numbered callouts with arrows point to specific elements: 1) A red box around the "Settings" link in the sidebar, with an arrow pointing from it to the "Settings" tab in the main content. 2) A red box around the "Template" tab in the main content, with an arrow pointing from it to the "ADD TEMPLATE" button. 3) A red box around the "Settings" link in the sidebar, with an arrow pointing from it to the "Settings" tab in the main content.

# Email Notifications – Event & Alarm Triggers

QMS Wep App

https://localhost:6443/settings

Home > Settings > Notification configuration

network-administrator

Add template step 1/1

Name

Subject

Content

ADD TEMPLATE

CREATE

4) Define a Name

5) Define Subject

6) Define Message Body with Variable Names

7) Click Create

Content can contains variable. The format to display a variable is {{theVariable}}. The available variables are: NodeName, SubSystem, TypeName, TypeCode, CategoryName, CategoryCode, Timestamp, Level, TriggerName and Message

# Email Notifications – Event & Alarm Triggers

SWISS  
QUANTUM+

The screenshot shows the ID Quantique web interface with a dark theme. On the left, a sidebar lists navigation options: Dashboard, Nodes, Paths, Monitoring, Firmware, Settings (which is highlighted in red), and Users. The main content area shows a breadcrumb path: Home > Settings > Notification configuration. A modal window titled "Add template step 1/1" is open. Inside the modal, there are three input fields: "Name" (QMS\_Alert\_Notification\_Email), "Subject" (QMS : ALERT - QKD), and "Content". The "Content" field contains a template string: "a problem has been detected on {{NodeName}}. On {{Timestamp}} an alarm {{TriggerName}} has been raised due to the event {{CategoryName}} : {{Message}}". Below the Content field, a note states: "Content can contain variables. The format to display a variable is {{theVariable}}. The available variables are: NodeName, SubSystem, TypeName, TypeCode, CategoryName, CategoryCode, Timestamp, Level, TriggerName and Message". At the bottom of the modal is a large green "CREATE" button. A cyan arrow points from a green box labeled "Example" on the left towards the "Content" field of the modal.

Example

Add template step 1/1

Name  
QMS\_Alert\_Notification\_Email

Subject  
QMS : ALERT - QKD

Content

a problem has been detected on {{NodeName}}. On {{Timestamp}}  
an alarm {{TriggerName}} has been raised due to the event  
{{CategoryName}} : {{Message}}

CREATE

# Create Distribution List



# Email Notifications – Event & Alarm Triggers

SWISS  
QUANTUM+

The screenshot shows the QMS Web App interface with a dark theme. On the left is a sidebar with navigation links: Dashboard, Nodes, Paths, Monitoring, Firmware, Settings (which is highlighted in red), and Users. The main content area has a breadcrumb path: Home > Settings > Notification configuration. The title 'Settings' is at the top, followed by tabs for 'Notification configuration', 'Maps', and 'TimeSeries'. Below these are sub-tabs: 'Notification configurations', 'Distribution list' (which is highlighted in blue), and 'Template'. A green callout box labeled '1) Click Settings' points to the 'Settings' link in the sidebar. Another green callout box labeled '2) Choose Add Distribution List' points to the 'ADD DISTRIBUTION LIST' button in the 'Distribution list' section. The 'Distribution list' section includes a search bar, a table with one row ('Chris'), and edit/delete icons. The bottom left corner of the sidebar displays the version 'V.1.0.2.0'.

# Email Notifications – Event & Alarm Triggers

SWISS  
QUANTUM+

QMS Wep App

https://localhost:6443/settings

Home > Settings > Notification configuration

English network-administrator

Add distribution list step 1/2

Distribution list name  
IT Management

3) Define Distribution Name

4) Click Next

NEXT

OD DISTRIBUTION LIST

V.1.0.2.0

# Email Notifications – Event & Alarm Triggers

SWISS  
QUANTUM+

QMS Wep App

https://localhost:6443/settings

Home > Settings > Notification configuration

English network-administrator

Add distribution list step 2/2

Dashboard

Nodes

Paths

Monitoring

Firmware

Settings

Users

ADD NEW EMAIL

EMAILS

5) Define Email Addresses

6) Click Add New Email

6) Click Create

PREVIOUS

CREATE

V1.0.2.0

# Add a Notification Trigger



# Email Notifications – Event & Alarm Triggers

The screenshot shows the QMS Wep App interface with a dark theme. On the left is a sidebar with navigation links: Dashboard, Nodes, Paths, Monitoring, Firmware, Settings (which is highlighted in red), and Users. The main content area has a breadcrumb path: Home > Settings > Notification configuration. Below this, there are tabs: Settings, Notification configurations (which is highlighted in red), Maps, and TimeSeries. A green arrow points from the text "1) Choose Settings" to the Settings tab. The main content area displays a table titled "Notification configurations" with columns: Name, Trigger count, Distribution list, and Template. A green box highlights the "Notification configurations" tab with the text "2) Choose Notification Configurations". At the bottom right of the table, there is a green button labeled "CREATE NEW NOTIFICATION CONFIGURATION". A green arrow points from the text "3) Click Create New Notification Configuration" to this button.

1) Choose Settings

2) Choose Notification Configurations

3) Click Create New Notification Configuration

# Email Notifications – Event & Alarm Triggers

SWISS  
QUANTUM+

QMS Wep App

https://localhost:6443/settings

Home > Settings > Notification configuration

English network-administrator

Add notification step 1/4

Name of the notification configuration

Name  
Low Key Rate

4) Create a meaningful name

5) Click Next

NEXT

0 of 0 pages (0 items)

V.1.0.2.0

# Email Notifications – Event & Alarm Triggers

SWISS  
QUANTUM+

6) Select Trigger, Such as Type Name

7) Choose Type such as Key Rate

8) Click Add Trigger

9) Click Next

Triggers selection

Select a trigger

Type name

Type name

QKD FPGA\_KEYRATE

Trigger

TypeNameTrigger

Details

Must have type name "QKD FPGA\_KEYRATE"

ADD TRIGGER

PREVIOUS

NEXT

# Email Notifications – Event & Alarm Triggers

SWISS  
QUANTUM+

QM Wep App

https://localhost:6443/settings

Home > Settings > Notification configuration

English network-administrator

Add notification step 3/4

Distribution list selection

Name

Chris

0 of 0 pages (0 items)

10) Choose Distribution List

11) Click Next

PREVIOUS

NEXT

V.1.0.2.0

# Email Notifications – Event & Alarm Triggers

SWISS  
QUANTUM+

QM Wep App

https://localhost:6443/settings

Home > Settings > Notification configuration

English network-administrator

Add notification step 3/4

Distribution list selection

Name

Chris

0 of 0 pages (0 items)

PREVIOUS

NEXT

10) Choose Distribution List

11) Click Next

# Modify docker-compose.override.yml



## Configure SMTP

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

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



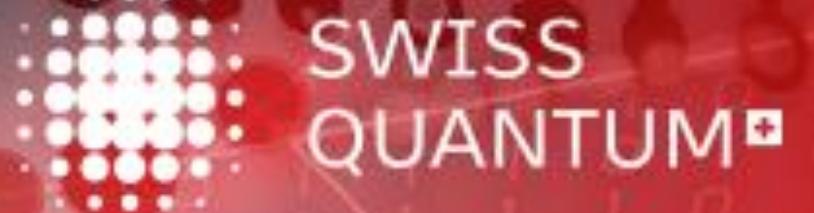
# Email Notifications – Event & Alarm Triggers

qms-backend:

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



# Email Notifications – Event & Alarm Triggers



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

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

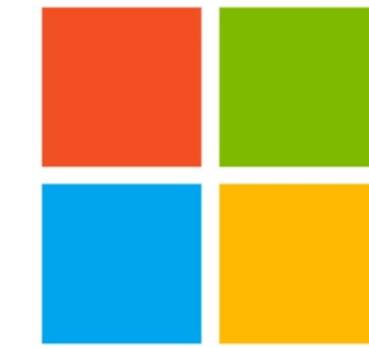
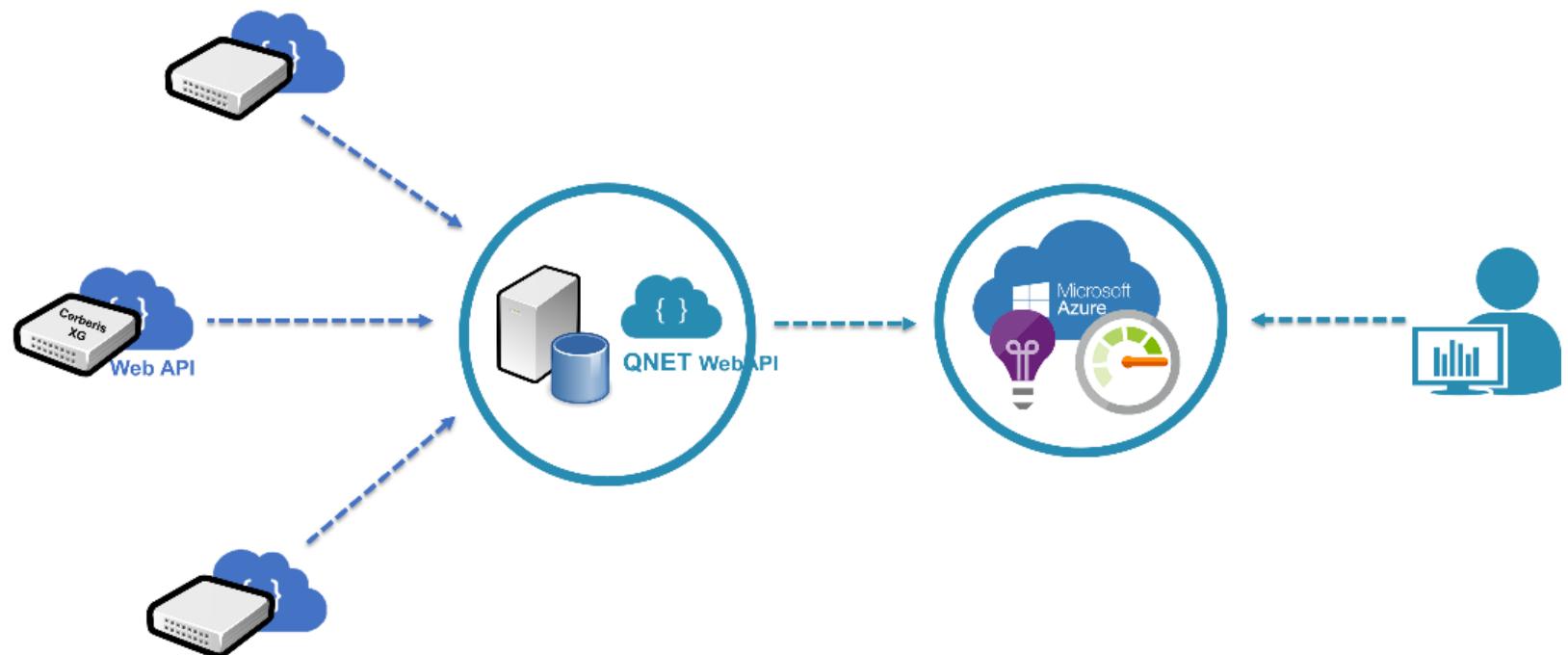
sudo docker-compose down  
sudo docker-compose up -d

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

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

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

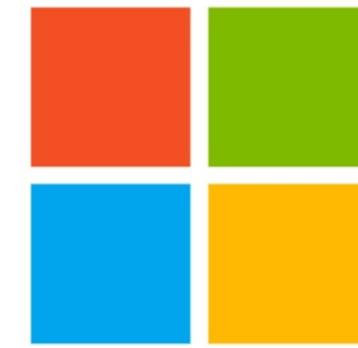
# MS Azure App Insights Integration



Microsoft  
Azure

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

See user guide or CLI help index for additional information



Microsoft  
Azure

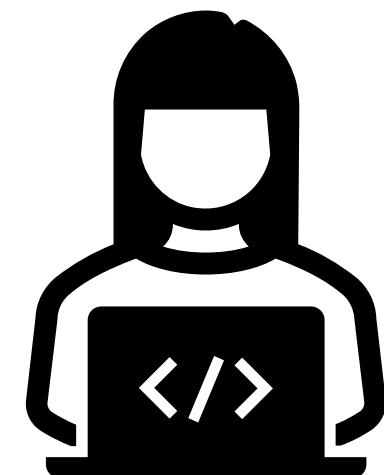
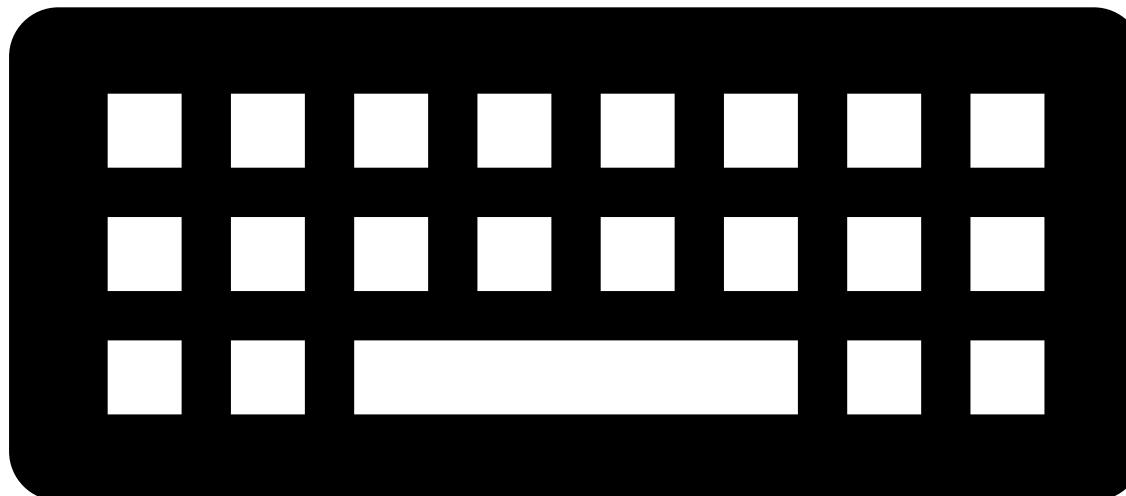
Azure AppInsights can also be enabled using QNET: See the following slides for additional QNET information.

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

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

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

# QNET Configuration API



- QMS enables central management where administrators can configure and monitor systems and operation remotely.
- QMS provides:
  - a Web UI console (Discussed earlier)
  - a QNET (configuration) WebApi
- QNET administration tool is supported on Linux x64 platforms and Microsoft Windows 10 x64. For both platforms an installer is provided.

- QNET is a command line interface providing all commands to manage your KMS/QKD network configuration. You can then:
  - Build the configuration of your KMS/QKD network by using these commands through scripts
  - Export QMS configuration into XML/Json files for one node or a group of nodes
  - (Re) import XML/Json configuration into QMS
  - Deploy configuration file directly into KMS nodes without the need of QMS
  - Collect current configuration directly from KMS nodes
- QNET relies on 2 new components:
  - QNET WebApi embedded with QMS
  - QNC WebApi embedded in KMS

# Sample QNet Commands (Script):

```
#!/bin/bash

# Authenticate First
qnet config gnefwebapi -url https://localhost:443 -user Admin -pwd Admin

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

#CREATE NODES
qnet create node CerberisGA Group1 -type Mgmt -ip 10.10.102 -uid Bd03d09736e4482aef743479208de20
qnet create node-network-interface CerberisGA Key 192.168.10.1
qnet create node-network-interface CerberisGA Kms 192.168.30.1

qnet create node CerberisGB Group1 -type Mgmt -ip 10.10.107 -uid Cf693ba73d456d1299428e998861
qnet create node-network-interface CerberisGB Key 192.168.10.2
qnet create node-network-interface CerberisGB Kms 192.168.30.2

qnet create node-link -algorithm BlockCipher KM5LNK CerberisGA CerberisGB

#CREATE PROVIDERS
## PROVIDER MGMT IP REQUIRED FOR CERBERISG-R
qnet create provider QDIA CerberisGA Cerberis3 -type Mgmt -ip 10.10.102
qnet create provider-vrf QDIA Any 192.168.20.1

## PROVIDER MGMT IP REQUIRED FOR CERBERISG-A
qnet create provider QDIB CerberisGB Cerberis3 -type Mgmt -ip 10.10.107
qnet create provider-vrf QDIB Any 192.168.20.2

qnet create provider QDIA CerberisGA Cerberis3 -type Any -ip 192.168.20.1
qnet create provider QDIB CerberisGB Cerberis3 -type Any -ip 192.168.20.2

qnet create provider-link PROVLINK QDIA QDIB --keyreq-interval 80

#CREATE CONSUMERS
qnet create consumer cisco CiscoCerberisGA -key-type QKEY -port 9999
qnet create consumer cisco CiscoCerberisGB -key-type QKEY -port 9999

qnet create consumer et5124 ET5124 CerberisGA -key-type QKEY -subject dn "CN=Et5124Server,L=Geneva,D=ID Quantique,O=QuantumSafe,CN=QDServer" -port 443
qnet create consumer et5124 ET5124 CerberisGB -key-type QKEY -subject dn "CN=Et5124Server,L=Geneva,D=ID Quantique,O=QuantumSafe,CN=QDServer" -port 443

qnet create consumer-link CiscoLINK CiscoC500 -pkc-ascii Cisco3234 -default-key-size 256
qnet create consumer-link ET512LINK ET512A ET512B -default-key-size 256

#CREATE PATHS
qnet create path CiscoC500 -mode Automatic
qnet create path ET512A ET512B -mode Automatic

#ADD CERTIFICATES
qnet add ca-cert consumer -ca-file ChrisCA.pem
qnet add cert QDServer.pem QDServer_pkcs12.pem
qnet deploy group Group1

qnet update consumer et5124 ET5124 CerberisGA -cert-file QDServer.pem --key-file QDServer_pkcs12.pem
qnet update consumer et5124 ET512B CerberisGB -cert-file QDServer.pem --key-file QDServer_pkcs12.pem

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



(Magnified on next slides)

Note that Clavis3 uses provider type “Cerberis3” and not “Clavis300”.

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

# Sample QNet Commands (Script):



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

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

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

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

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

# Sample QNet Commands (Script):



#CREATE PROVIDERS

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

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

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

#CREATE CONSUMERS

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

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

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

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

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

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

# Sample QNet Commands (Script):



```
#CREATE PATHS
```

```
qnet create path CiscoA CiscoB -mode Automatic  
qnet create path ETSIA ETSIB -mode Automatic
```

```
#ADD CERTIFICATES
```

```
qnet add ca-cert consumer --ca-file ChrisCA.pem  
qnet add cert QKDServer.pem QKDServer_pkey.pem
```

```
qnet deploy group Group1
```

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

```
#DEPLOY THE FINAL CONFIG TO ALL NODES
```

```
qnet deploy group Group1
```

# Check if the system is working

## Understand QKD logs



# QKD logs

```
admin> help  
1.1.32
```

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

Usage: qnet [command] [options]

Options:

- version Show version information.
- help Show help information.

Commands:

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

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

```
admin>
```

**logtail**  
**logcp**

**Monitor the system log**

→ Provide real-time view on the logs

**Copy the system logs**

→ Copy logs on USB stick

# QKD logs – logtail command



Logtail's command

```
admin> logtail qkd
```

“live” logs

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

# QKD logs – Understand the logs

- Thousands of log lines are present, but you can easily find the information that you need.
- **Alice and Bob logs are different, they don't show the same type of information**
- You can use logtail <log> --pattern to filter by type of information

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

Usage: logtail [options] <log>

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

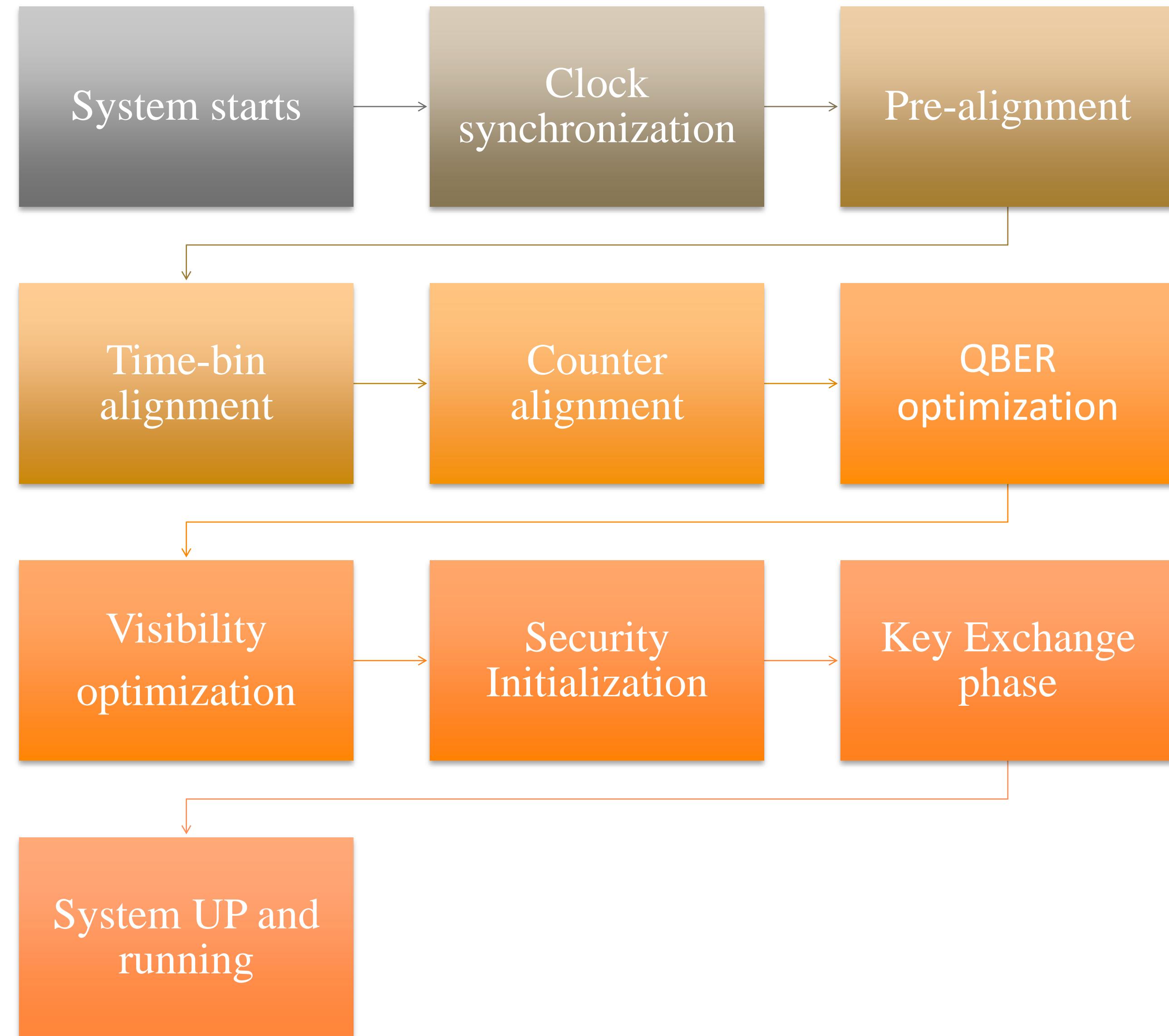
Options:
  --help        Show help information.
  -p|--pattern <PATTERNS> Monitor the system log with a specific pattern

admin>
```

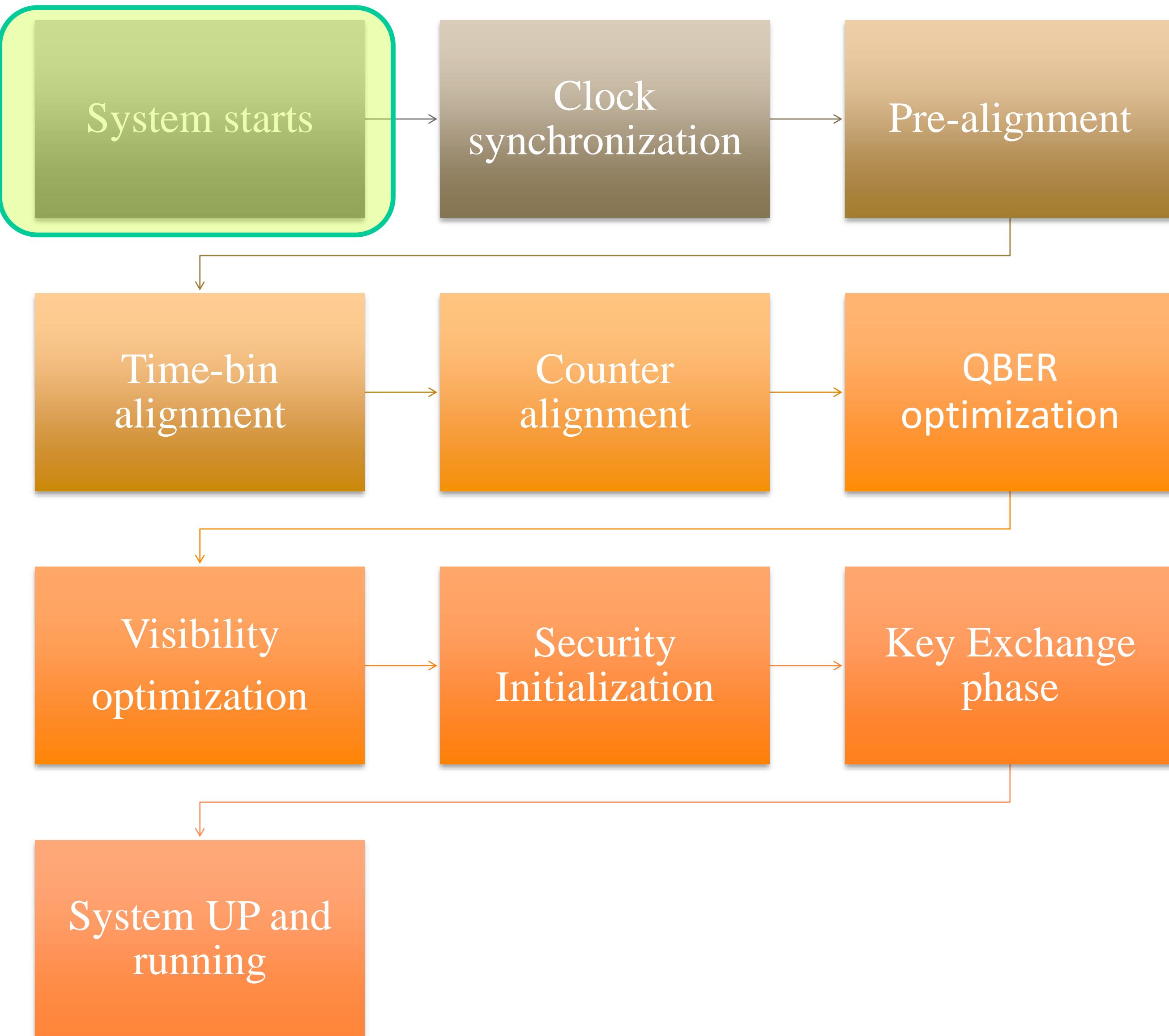
- Logs lines can be either **INFO** or **DEBUG**

- Before understanding the logs, we need to **understand the different steps** that are done during the system alignment phase
- If **one step is not properly finished** the system will not go to the **next step**
- Note there is a separate document titled “**Understanding QKD Log Files**” Please refer to this document for additional information.

# QKD logs – Understand the QKD processes



# QKD logs – Understands the QKD processes



# QKD logs – Understands the QKD processes

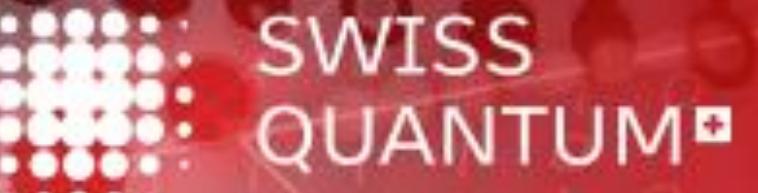
```
=====
Communicator Alice/Bob
2.0.0.26678 RELEASE                                     Alice or Bob QKD Blade software version
Copyright (C) 2015-2021 ID Quantique SA

=====
IDQ4P 1.4.0                                              IDQ4P version
ZeroMQ 4.3.4                                            ZeroMQ version
MsgPack 1.1.0                                             MessagePack version
=====
```

Subsequently, the log files show general system information:

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

# QKD logs – Understands the QKD processes



This is followed by general FPGA settings.

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

# QKD logs – Understands the QKD processes



Then, we find the general optical settings.

On Alice:

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

On Bob:

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

# QKD logs – Understands the QKD processes



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

On Alice:

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

# QKD logs – Understands the QKD processes



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

On Bob:

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

# QKD logs – Understands the QKD processes

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

On Alice (pre-alignment) :

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

On Bob:

Timebin Alignment:	
Trigger cycles:	31250000
Min detections:	1000
Detector 1 threshold:	0.9
Detector 2 threshold:	0.9
Counter Alignment:	
Detector 1 max bit address:	21
Detector 2 max bit address:	8
Max counts:	1024
Threshold:	0.4

# QKD logs – Understands the QKD processes

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

QBER Minimization:	
Trigger cycles:	31250000
Min detections:	1000
DATA delay step:	1.1e-08
IM bias voltage step:	0.05
Visibility Maximization:	
Trigger cycles:	31250000
Min detections:	1000
MONITOR delay step:	1.1e-08
Coarse laser current min:	-0.0015
Coarse laser current max:	0.0015
Coarse laser current step:	0.0001
Fine laser current min:	-0.0001
Fine laser current max:	0.0001
Fine laser current step:	1e-05

# QKD logs – Understands the QKD processes



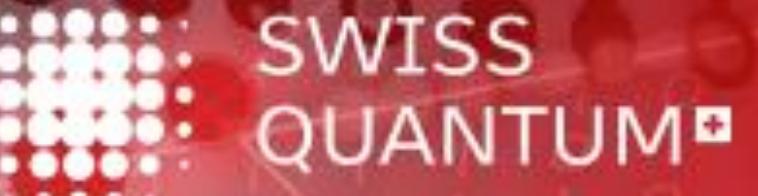
On both Alice and Bob, we specify the timeout if the system generates no keys:

Key Exchange:

Timeout: 20 min

Time with no key generation after which the system restarts

# QKD logs – Understands the QKD processes



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

Optics Regulation:	
DATA dark counts:	694
DATA deadtime:	50
DATA detection efficiency:	0.2
MONITOR dark counts:	691
MONITOR deadtime:	50
MONITOR detection efficiency:	0.2
Bob's coupler ratio:	0.8
QBER integration time:	10
QBER epsilon:	0
Visibility integration time:	10
Visibility epsilon:	0
Modulator bias reduced step:	0.1
Modulator epsilon:	0.3
Laser current min:	0.045
Laser current max:	0.055
Laser current period:	0.001
Laser current jump multiplier:	5

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

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

# QKD logs – Understands the QKD processes



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

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

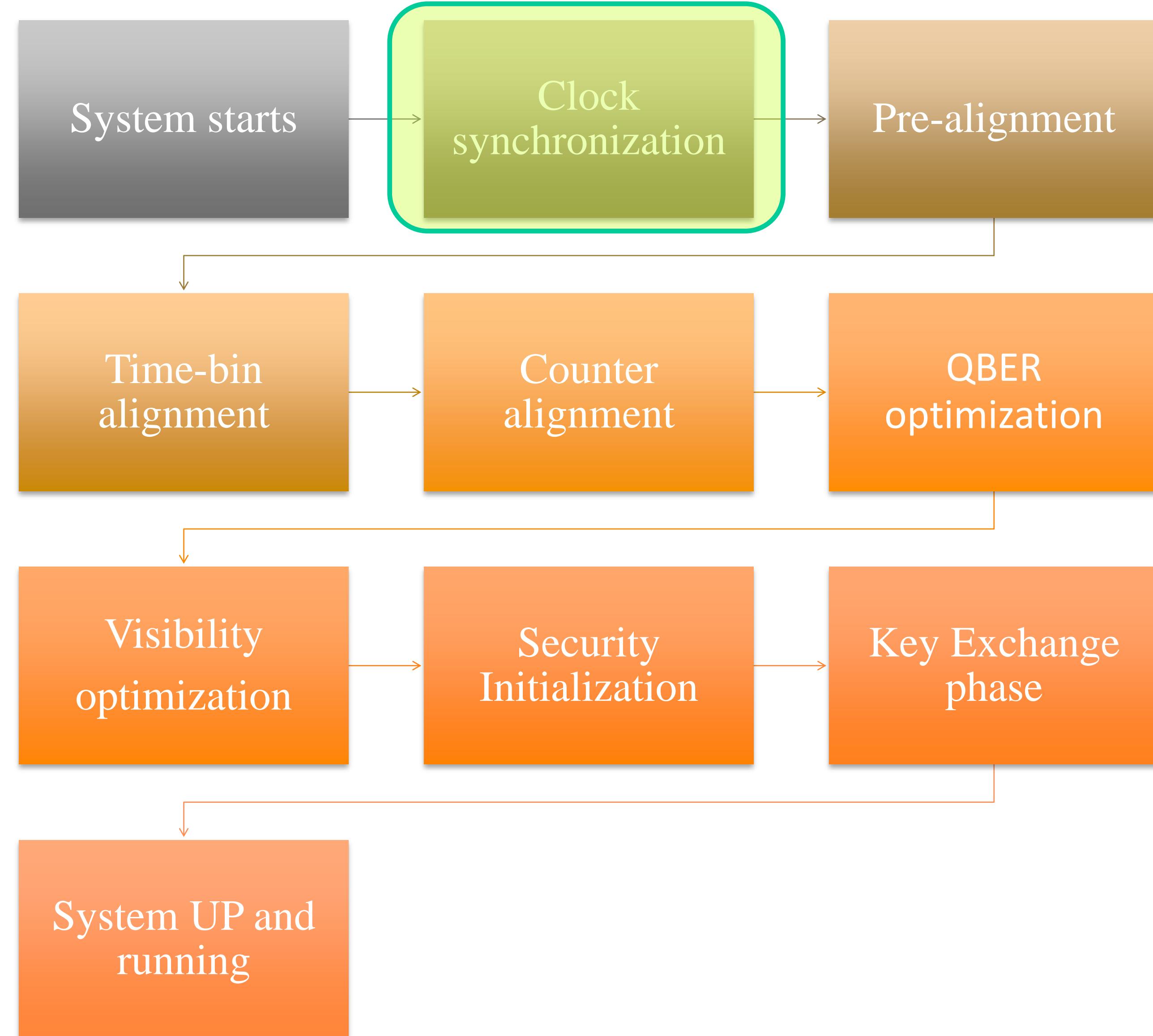
On Alice :

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

On Bob:

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

# QKD logs – Understands the QKD processes



## Clock synchronization

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

```
Local FPGA state: SERVICE_CHANNEL_SYNC
Remote FPGA state: SERVICE_CHANNEL_SYNC
Service channel synchronization done: false
Local FPGA state: SERVICE_CHANNEL_SYNC
Remote FPGA state: SERVICE_CHANNEL_SYNC
Waiting for clock recovery state...
Service channel synchronization done: true
Local FPGA state: IDLE
```

Current state of the local FPGA

Current state of the remote FPGA

Indicates asynchronous FPGA clocks

Indicates locking of the local PLL

FPGA clocks synchronized

# QKD logs – Understands the QKD processes

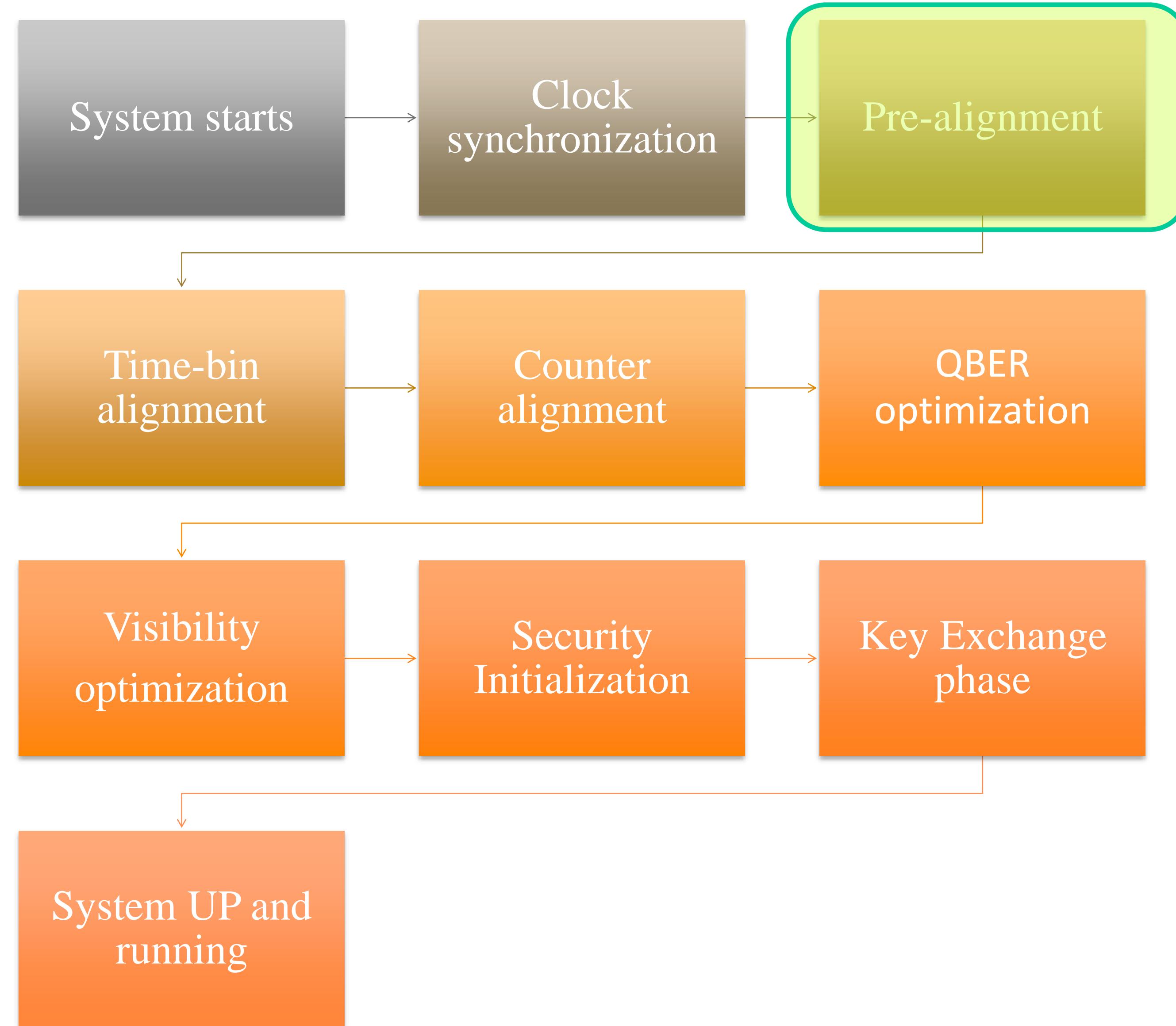


- If the FPGA clock is synchronized the system goes to the next step

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

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

# QKD logs – Understands the QKD processes



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

## *Pre-Alignment*

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

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

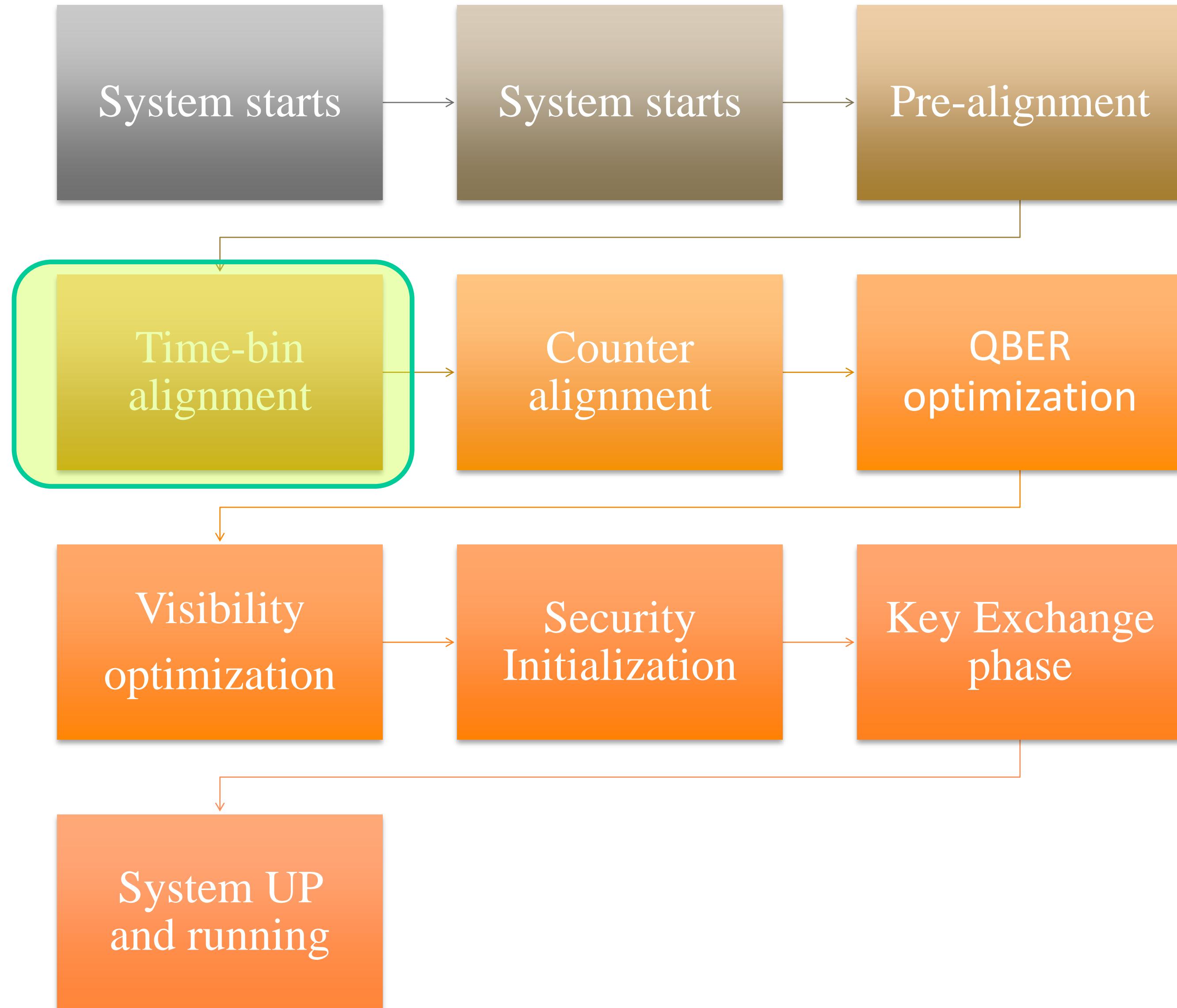
Coarse scan

# QKD logs – Understands the QKD processes



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

# QKD logs – Understands the QKD processes



# QKD logs – Understands the QKD processes

## *Time-bin alignment*

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

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

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

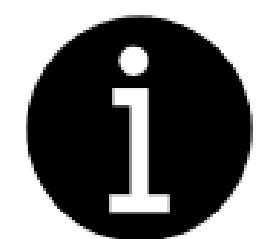
...
4 Aligning MONITOR timebin: found best alignment for delay = 2330, shift = 13, correlation =
0.997888.
```

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



#### NOTE

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

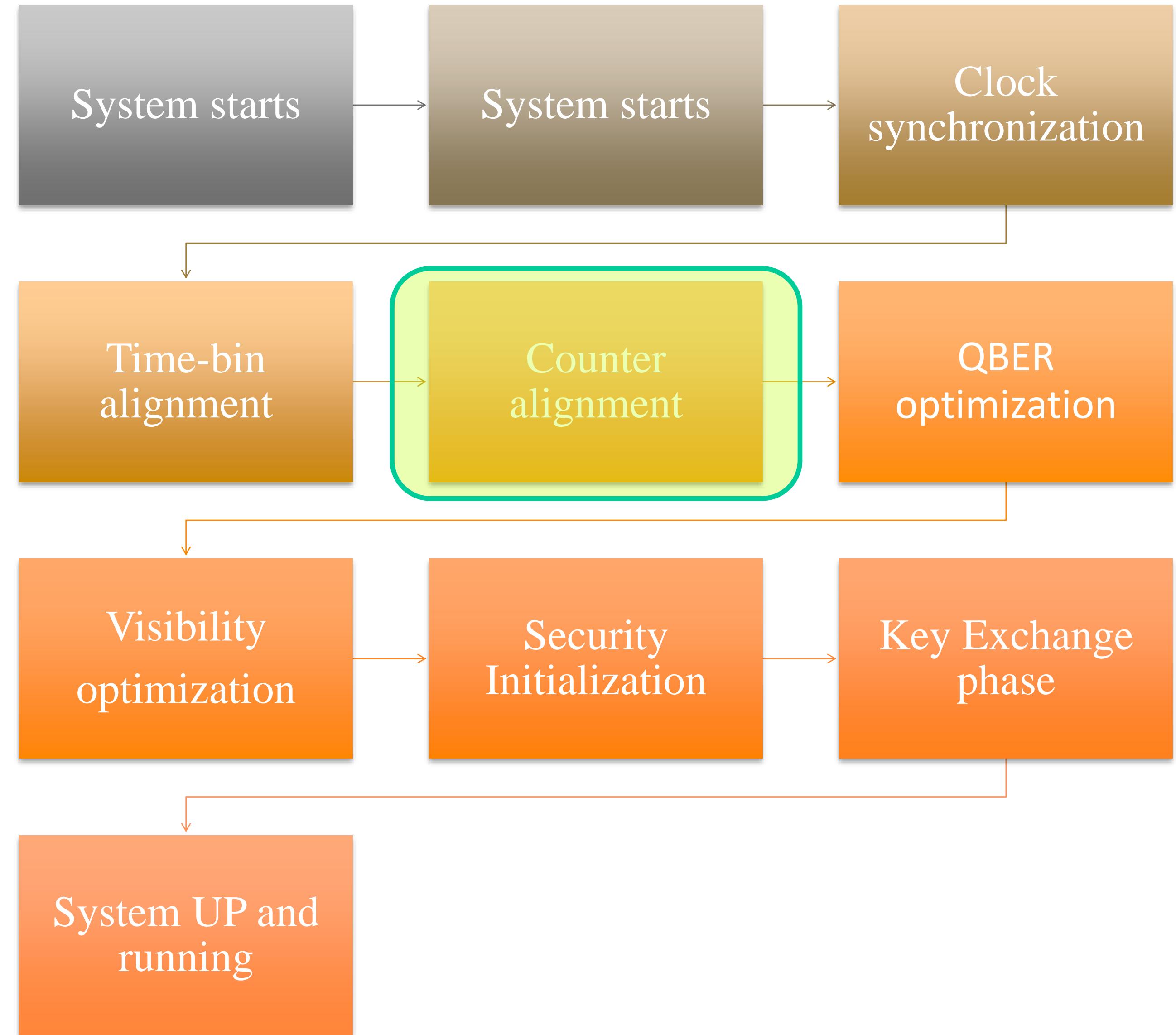


## NOTE

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

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

# QKD logs – Understands the QKD processes



# QKD logs – Understands the QKD processes

```
Aligning DATA counter...

Aligning counter: alignment at bit 0

Aligning counter: XOR Counter 0 = 998, XOR Counter 1 = 26, correlation = 0.974609, QBER =
0.0253906

...
Aligning counter: alignment at bit 20
1 Aligning counter: XOR Counter 0 = 26, XOR Counter 1 = 998, correlation = 0.0253906, QBER =
0.974609

...
Aligning counter: bitflip bit 20
1 Aligning counter: XOR Counter 0 = 1002, XOR Counter 1 = 22, correlation = 0.978516, QBER =
0.0214844

DATA counter aligned.

Aligning MONITOR counter...
Aligning counter: alignment at bit 0
2 Aligning counter: XOR Counter 0 = 74, XOR Counter 1 = 950, correlation = 0.0722656, QBER =
0.927734

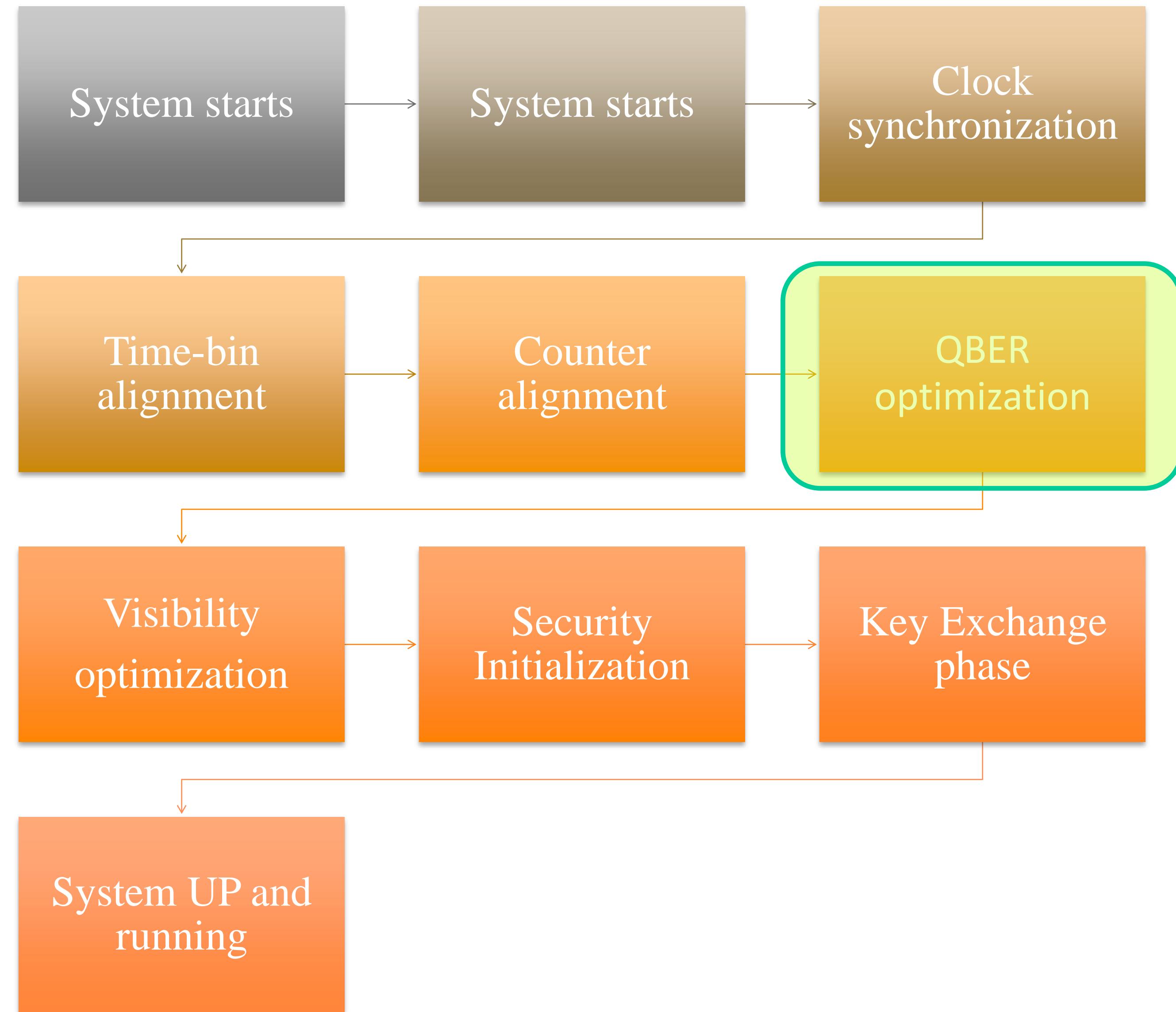
Aligning counter: bit 0 needs a bitflip.
Aligning counter: adding 1 Qbit delay.
Aligning counter: adding QbitDelay*4bit=4 bit delay.
3 Aligning counter: current clock delay = 0, shift delay = 13
Aligning counter: adding 4 bit delay
Aligning counter: new clock delay = 0, shift delay = 1
4 Aligning counter: XOR Counter 0 = 973, XOR Counter 1 = 51, correlation = 0.950195, QBER =
0.0498047

...
Aligning counter: alignment at bit 7
1 Aligning counter: XOR Counter 0 = 892, XOR Counter 1 = 132, correlation = 0.871094, QBER =
0.128906

MONITOR counter aligned.
```

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

# QKD logs – Understands the QKD processes



# QKD logs – Understands the QKD processes

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

On Alice:

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

Currently applied modulator bias voltage

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

On Bob:

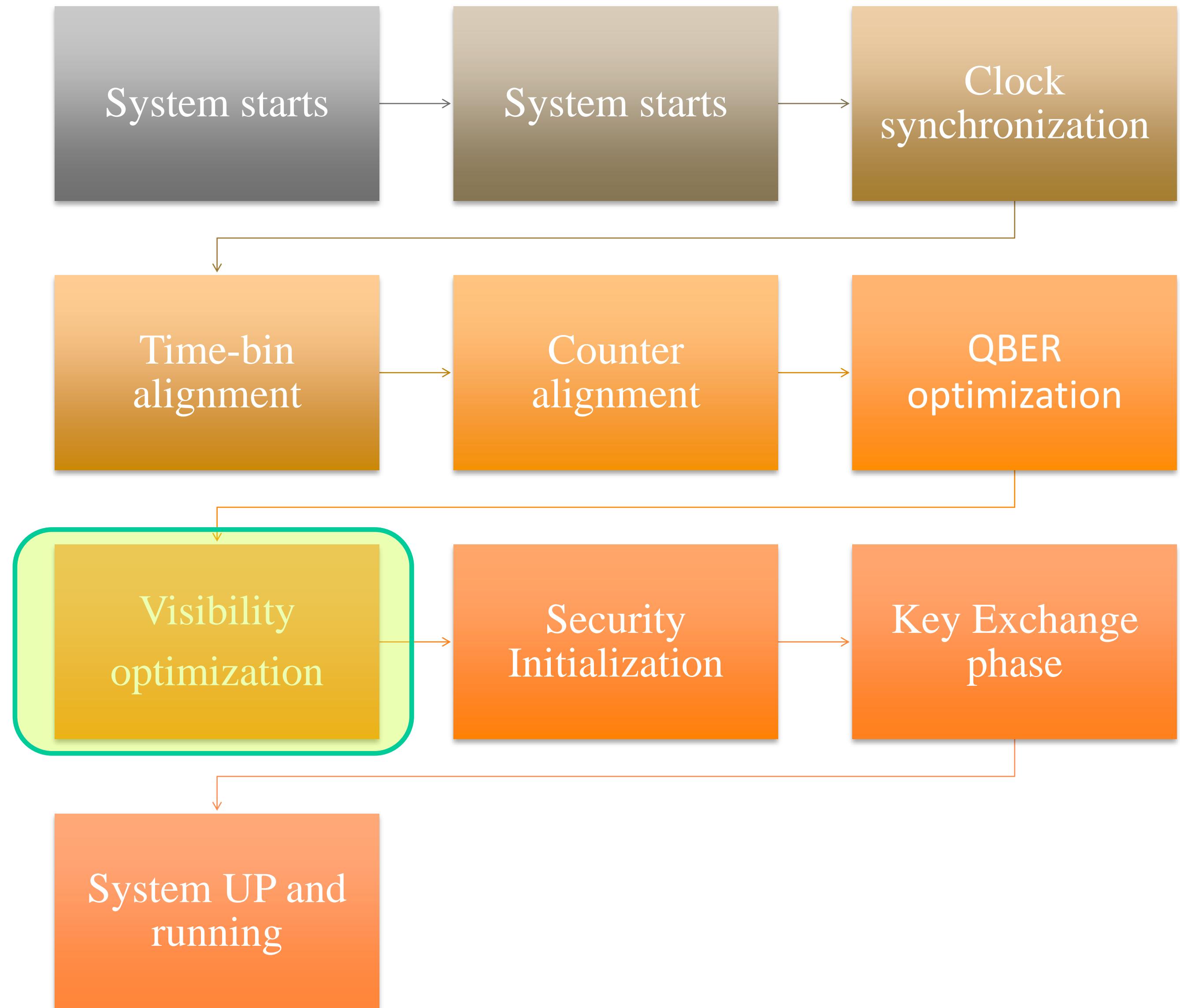
```
Minimizing QBER...
Minimizing QBER: modulator bias = 6.11261 V.
Minimizing QBER: QBER = 0.0159744.
...
Minimizing QBER: QBER = 0.0162947.
Minimizing QBER: DATA delay = 2156 ps.
```

Currently applied modulator bias voltage and  
corresponding measured QBER

Currently applied DATA delay

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

# 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. > 0.95 is a good value)

Alice

```
Maximizing visibility...
Maximizing visibility: current = 44999 uA.
Maximizing visibility: current = 43499 uA.
:
Maximizing visibility: current = 44639 uA.
Visibility maximized.
Leaving ExecutingGeneralInitialization state.
```

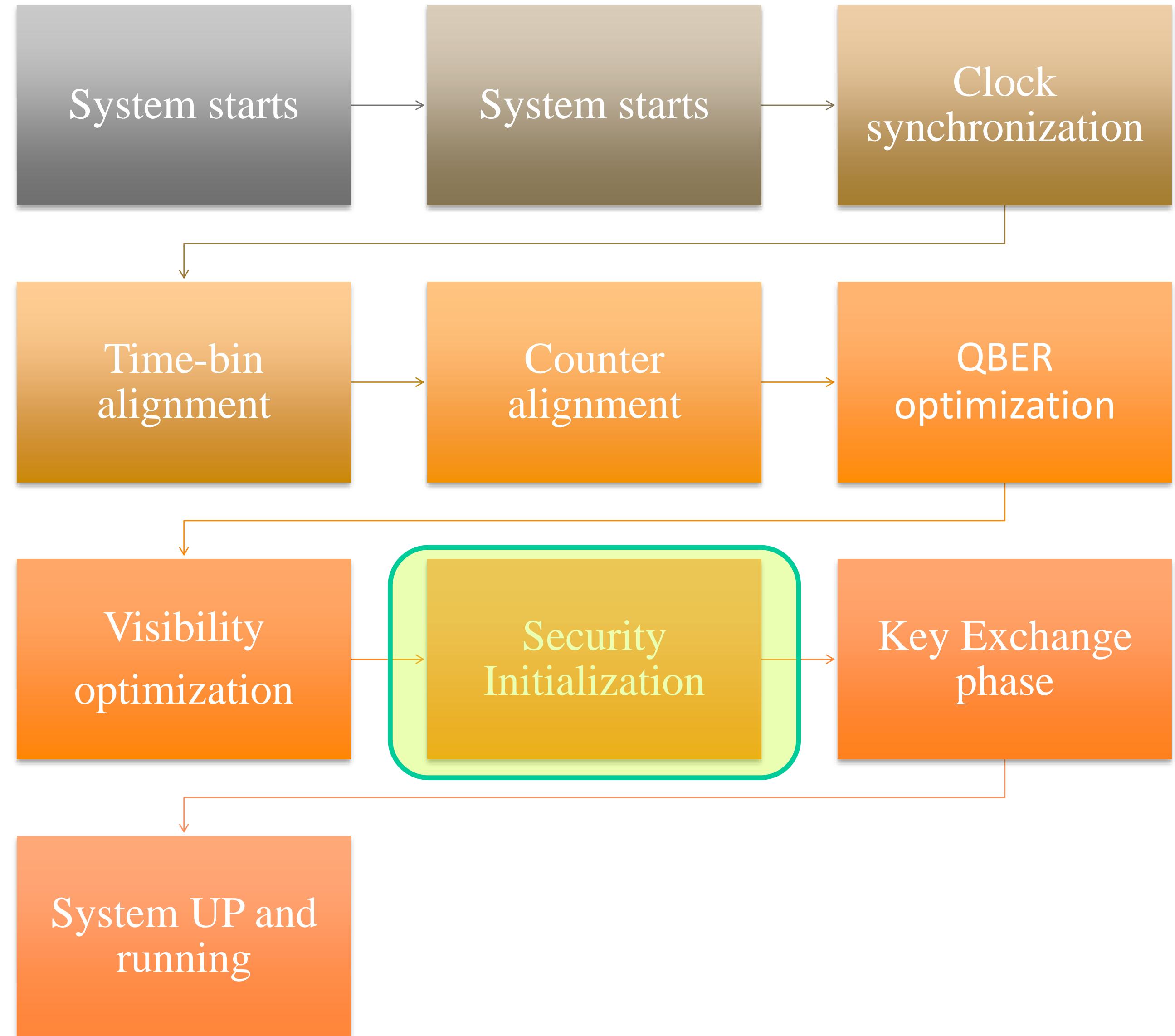
Starting the Visibility Optimization step.  
Currently applied laser bias current.  
Finalizing the Visibility Optimization.  
Finalizing the General Initialization phase.

Bob

```
Maximizing visibility...
Remote FPGA Optics Optimization state: 1 (expected 2). Waiting for Alice.
Maximizing visibility: changing laser current value...
Maximizing visibility: current = 44999 uA.
Maximizing visibility: current = 43499 uA.
Maximizing visibility: visibility = 0.467572.
:
Maximizing visibility: current = 45999 uA.
Maximizing visibility: visibility = 0.966131.
Maximizing visibility: found highest visibility for ...
Visibility maximized.
Leaving ExecutingGeneralInitialization state.
```

Starting the Visibility Optimization step.  
Currently applied laser bias current and corresponding measured Visibility.  
Finalizing the Visibility Optimization.  
Finalizing the General Initialization phase.

# QKD logs – Understands the QKD processes



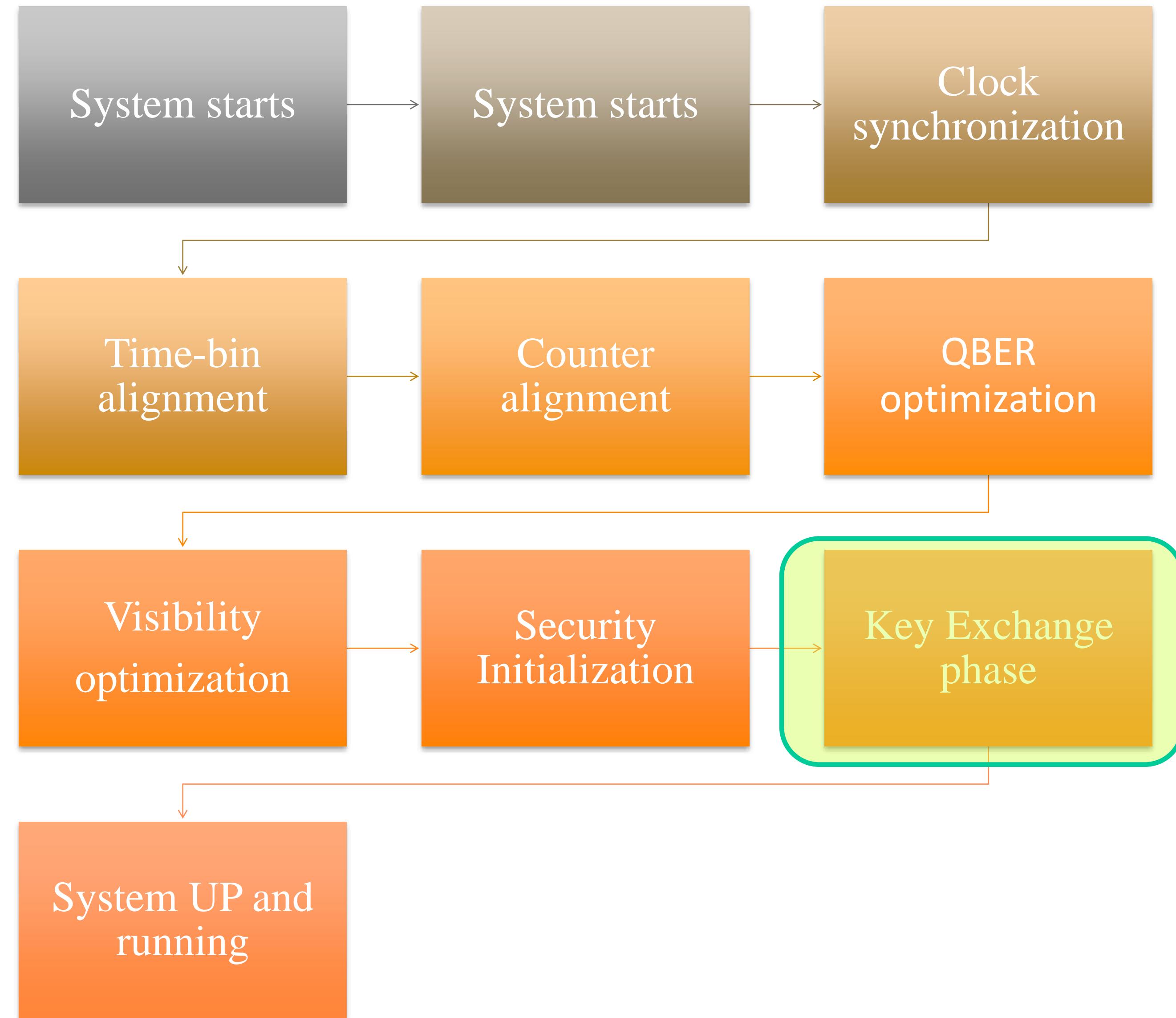
# QKD logs – Understands the QKD processes



- If the initial key is not provided, you should verify the QMS configuration has been deployed.
- If needed you could re-create and re-deploy the path.

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

# QKD logs – Understands the QKD processes



# QKD logs – Understands the QKD processes

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

Alice

```
Entering Running state.  
Key exchange started.  
QBER: 0.021227  
Visibility: 0.966967  
  
...  
Regulating QBER: modulator bias = 6.63949 V, QBER = 0.0119892.  
Regulating visibility: laser current = 0.044639 A, visibility = 0.958673.  
...  
Updating PA parameters: FullBlockHashValids: 512, FullBlockHashErrors:1  
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  
Current compression ratio: 0.0952289  
...  
Starting the Key Exchange.  
Key exchange started.  
Currently measured QBER.  
Currently measured Visibility.  
  
QBER regulation.  
Visibility regulation.  
  
Hash errors.  
  
New compression ratio for next key.  
Currently applied compression ratio.
```

Bob

```
Entering Running state.  
Starting key exchange...  
Key rate: 0 bits/s  
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  
Current compression ratio: 0.0952289  
Key rate: 2227.2 bits/s  
...  
  
Starting the Key Exchange.  
Key exchange started.  
Current secret key rate.  
  
New compression ratio for next key.  
Currently applied compression ratio.  
Current secret key rate.
```

# CerberisXG/Cerberis3



## SNMP monitoring

# Management and monitoring



- CerberisXG/QKD and QNC parameters and realtime values can be checked using:
  - SNMPv3 walk
  - Syslogs
  - Logs
  - QMS
- The recommended lab tool to use snmp walk is SNMPB (open source)



# SNMPB configuration

## 1. Download and Install SNMPB

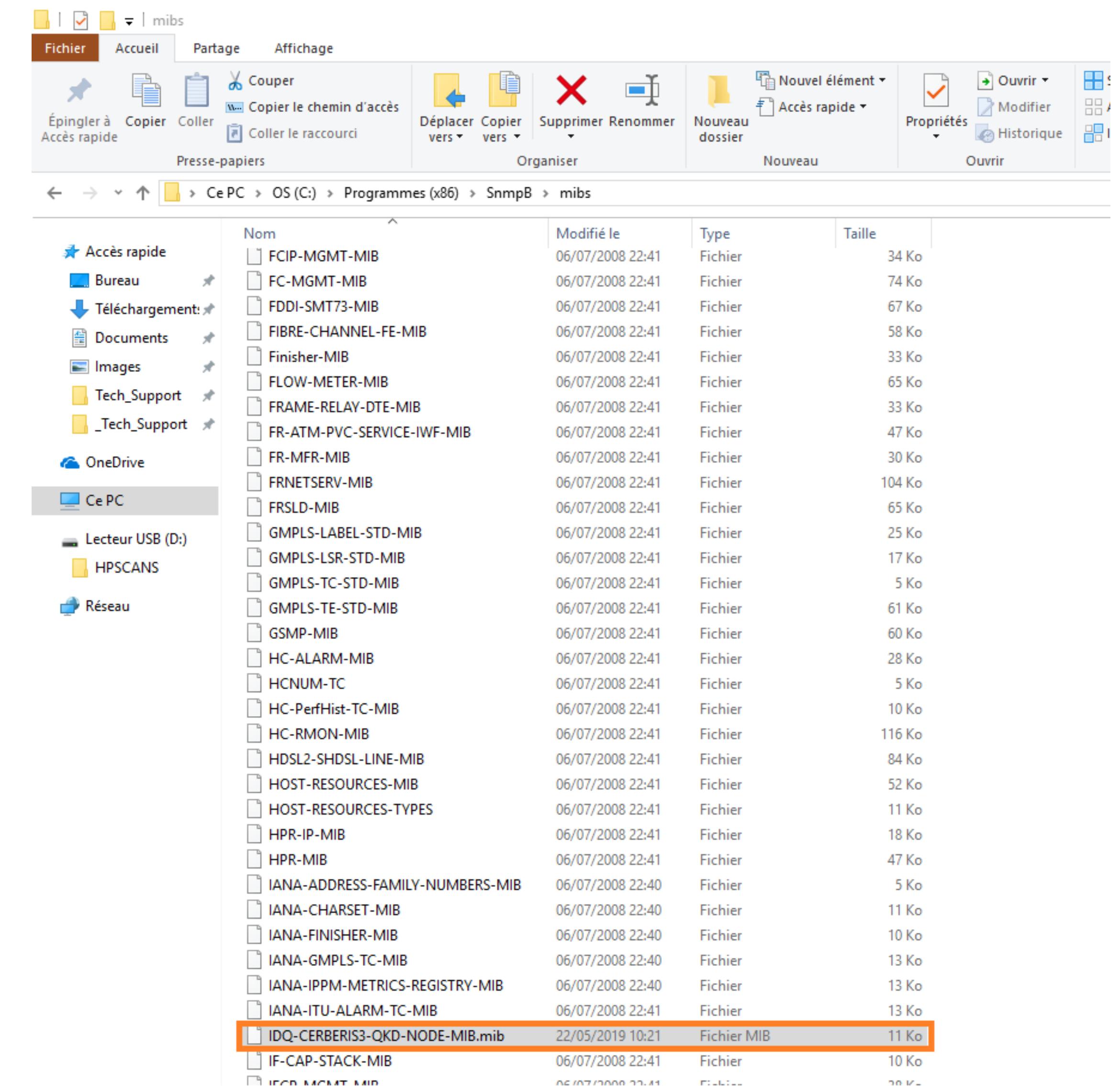
- Go to <https://sourceforge.net/projects/snmpb/>
- Download the file
- Install the tool

The screenshot shows the SourceForge project page for SnmpB. At the top, there's a navigation bar with links for Open Source Software, Business Software, Services, and Resources. Below the navigation is a banner for InterDiscount with a 'BON 80.-' offer and a Huawei P Smart (2019) phone for 169.-. The main content area has a dark background. It features a thumbnail of the SnmpB application, its status as 'Beta', and credits to 'martj, ulidtko'. Below this, there are statistics: 21 reviews (with a 5-star rating), 508 downloads this week, and a last update date of 2019-05-13. A large green 'Download' button is prominently displayed, outlined in red. Below the download button are links for BSD, Windows, Mac, and Linux. A navigation menu at the bottom includes Summary, Files, Reviews, Support, Code, Wiki, Discussion, Tickets, and News. A descriptive text block explains that SnmpB is an SNMP MIB browser written in QT, supporting SNMPv1, SNMPv2c & SNMPv3, and can browse/edit/load/add MIB files, query SNMP agents, support agent discovery, trap events, and graph plotting. A 'Features' section lists: • SNMP browser, • MIB editor, • SNMPv3 support, • SNMP traps receiving, and • SNMP agent discovery. At the very bottom, there's a 'Project Samples' section.

# 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 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”

The screenshot shows the SNMPB software interface. At the top, there is a menu bar with File, Tools, Options, Help. Below it is a toolbar with tabs: Tree, Modules, Editor, Discovery, Traps, Graphs, Log. The main window has two main sections: "Available MIB modules" on the left and "Loaded MIB modules" on the right.

**Available MIB modules:** A list of MIB modules. One module, "IDQ-CERBERIS3-QKD-NODE-MIB.mib", is highlighted with a red box and has a large orange arrow pointing from the "Available" list to the "Loaded" list. In the "Loaded MIB modules" section, there is a blue button with a right-pointing arrow (">") and a red circle around it, indicating the action to move the selected module.

**Loaded MIB modules:** A table showing the currently loaded MIB modules. The table includes columns for Module, Required, Language, and Path.

Module	Required	Language	Path
IF-MIB	no	SMIv2	mibs\IF-MIB
SNMPv2-SMI	no	SMIv2	mibs\SNMPv2-SMI
SNMPv2-TC	no	SMIv2	mibs\SNMPv2-TC
SNMPv2-CONF	no	SMIv2	mibs\SNMPv2-CONF
SNMPv2-MIB	no	SMIv2	mibs\SNMPv2-MIB
RFC1213-MIB	no	SMIv1	mibs\RFC1213-MIB
RFC1155-SMI	no	SMIv1	mibs\RFC1155-SMI
RFC-1212	no	SMIv1	mibs\RFC-1212
SNMP-FRAMEWORK-MIB	no	SMIv2	mibs\SNMP-FRAMEWORK-MIB
SNMP-NOTIFICATION-MIB	no	SMIv2	mibs\SNMP-NOTIFICATION-MIB
SNMP-TARGET-MIB	no	SMIv2	mibs\SNMP-TARGET-MIB
SNMPv2-TM	no	SMIv2	mibs\SNMPv2-TM
SNMP-VIEW-BASED-ACM-MIB	no	SMIv2	mibs\SNMP-VIEW-BASED-ACM-MIB

**Module info:** A detailed view of the selected MIB module, "IDQ-CERBERIS3-QKD-NODE-MIB". The "Requires:" field contains "SNMPv2-TC", "SNMPv2-CONF", and "SNMPv2-SMI", which are highlighted with a green box. A large orange arrow points from the "Available" list to the "Loaded" list, indicating the direction of the move.

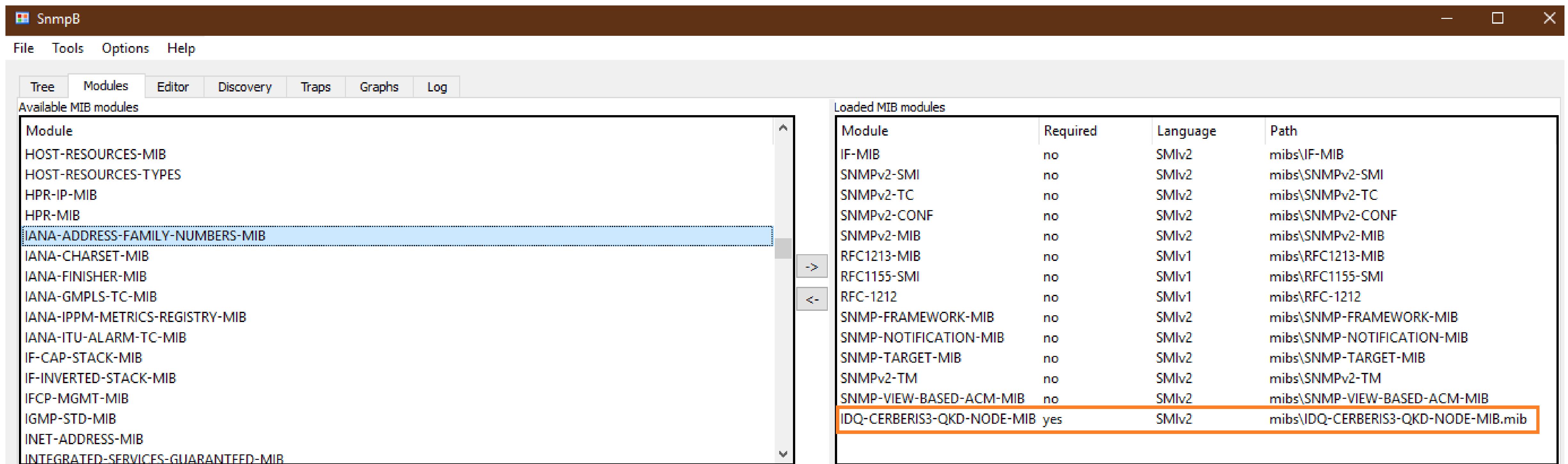
**Text overlay:** "Wrong information. Do not pay attention" is displayed in the bottom left corner of the "Module info" section, with a green box highlighting the text.

**Callout box:** A green callout box on the right side of the screen contains the text "Move Both MIB Files!".

# SNMPB configuration

## 3. Configure MIBs on SNMPB

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



The screenshot shows the SnmpB application window. The top menu bar includes File, Tools, Options, Help, Tree, Modules, Editor, Discovery, Traps, Graphs, and Log. The main interface is divided into two main sections: 'Available MIB modules' on the left and 'Loaded MIB modules' on the right.

**Available MIB modules:**

- HOST-RESOURCES-MIB
- HOST-RESOURCES-TYPES
- HPR-IP-MIB
- HPR-MIB
- IANA-ADDRESS-FAMILY-NUMBERS-MIB
- IANA-CHARSET-MIB
- IANA-FINISHER-MIB
- IANA-GMPLS-TC-MIB
- IANA-IPPM-METRICS-REGISTRY-MIB
- IANA-ITU-ALARM-TC-MIB
- IF-CAP-STACK-MIB
- IF-INVERTED-STACK-MIB
- IFCP-MGMT-MIB
- IGMP-STD-MIB
- INET-ADDRESS-MIB
- INTEGRATED-SERVICES-GUARANTEED-MIB

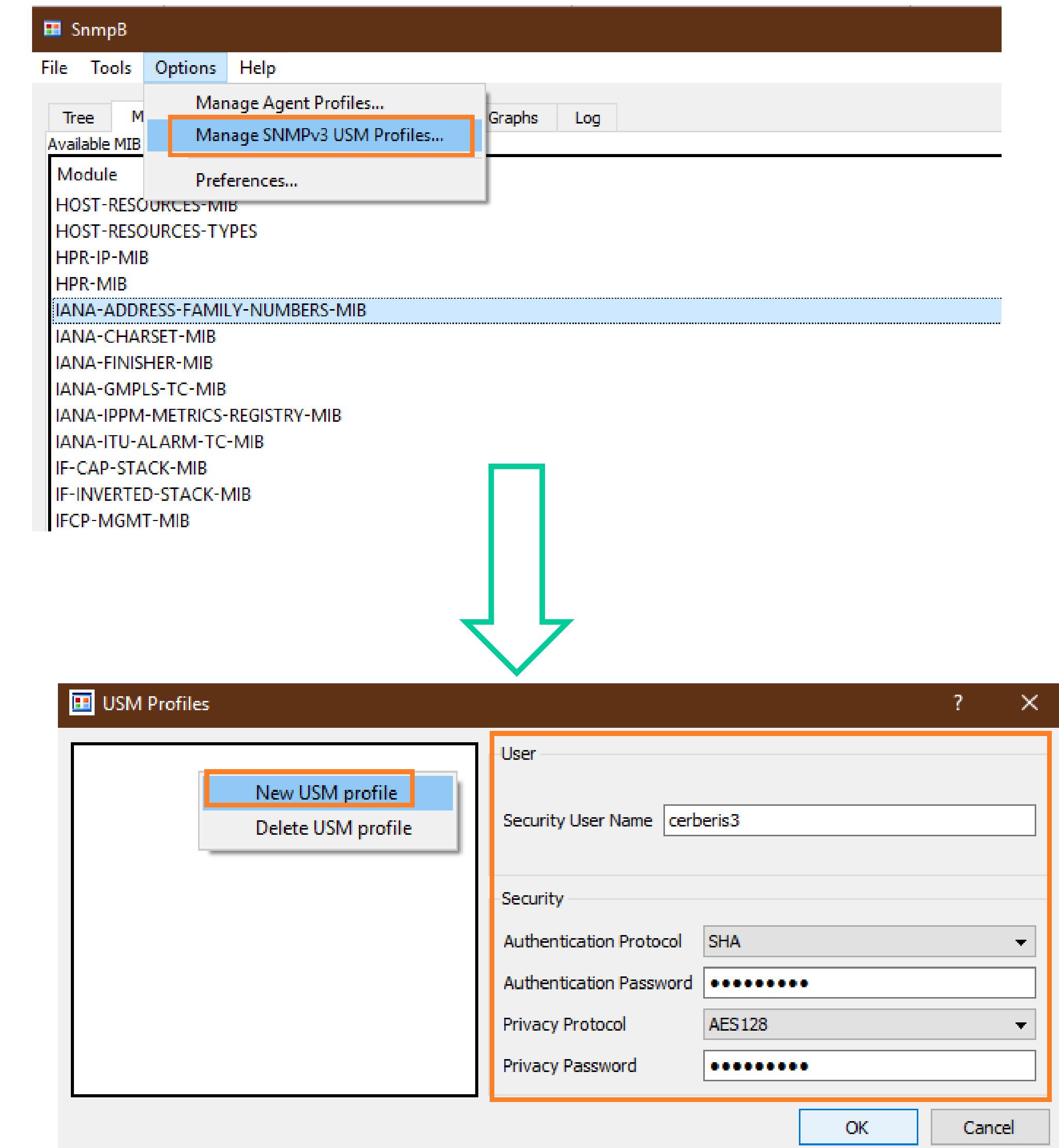
**Loaded MIB modules:**

Module	Required	Language	Path
IF-MIB	no	SMLv2	mibs\IF-MIB
SNMPv2-SMI	no	SMLv2	mibs\SNMPv2-SMI
SNMPv2-TC	no	SMLv2	mibs\SNMPv2-TC
SNMPv2-CONF	no	SMLv2	mibs\SNMPv2-CONF
SNMPv2-MIB	no	SMLv2	mibs\SNMPv2-MIB
RFC1213-MIB	no	SMLv1	mibs\RFC1213-MIB
RFC1155-SMI	no	SMLv1	mibs\RFC1155-SMI
RFC-1212	no	SMLv1	mibs\RFC-1212
SNMP-FRAMEWORK-MIB	no	SMLv2	mibs\SNMP-FRAMEWORK-MIB
SNMP-NOTIFICATION-MIB	no	SMLv2	mibs\SNMP-NOTIFICATION-MIB
SNMP-TARGET-MIB	no	SMLv2	mibs\SNMP-TARGET-MIB
SNMPv2-TM	no	SMLv2	mibs\SNMPv2-TM
SNMP-VIEW-BASED-ACM-MIB	no	SMLv2	mibs\SNMP-VIEW-BASED-ACM-MIB
IDQ-CERBERIS3-QKD-NODE-MIB	yes	SMLv2	mibs\IDQ-CERBERIS3-QKD-NODE-MIB.mib

# SNMPB configuration

## 4. Create SNMPv3 user

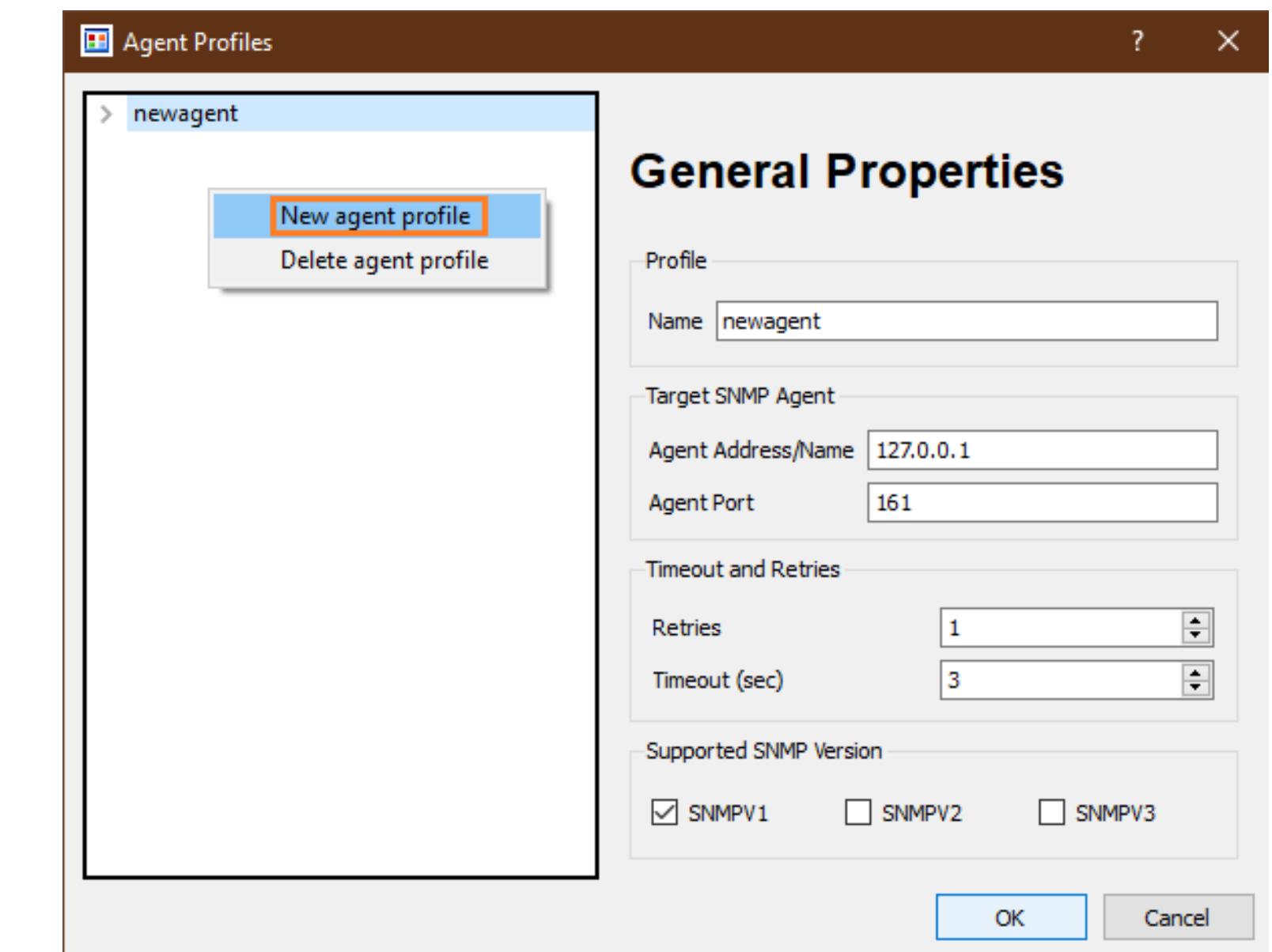
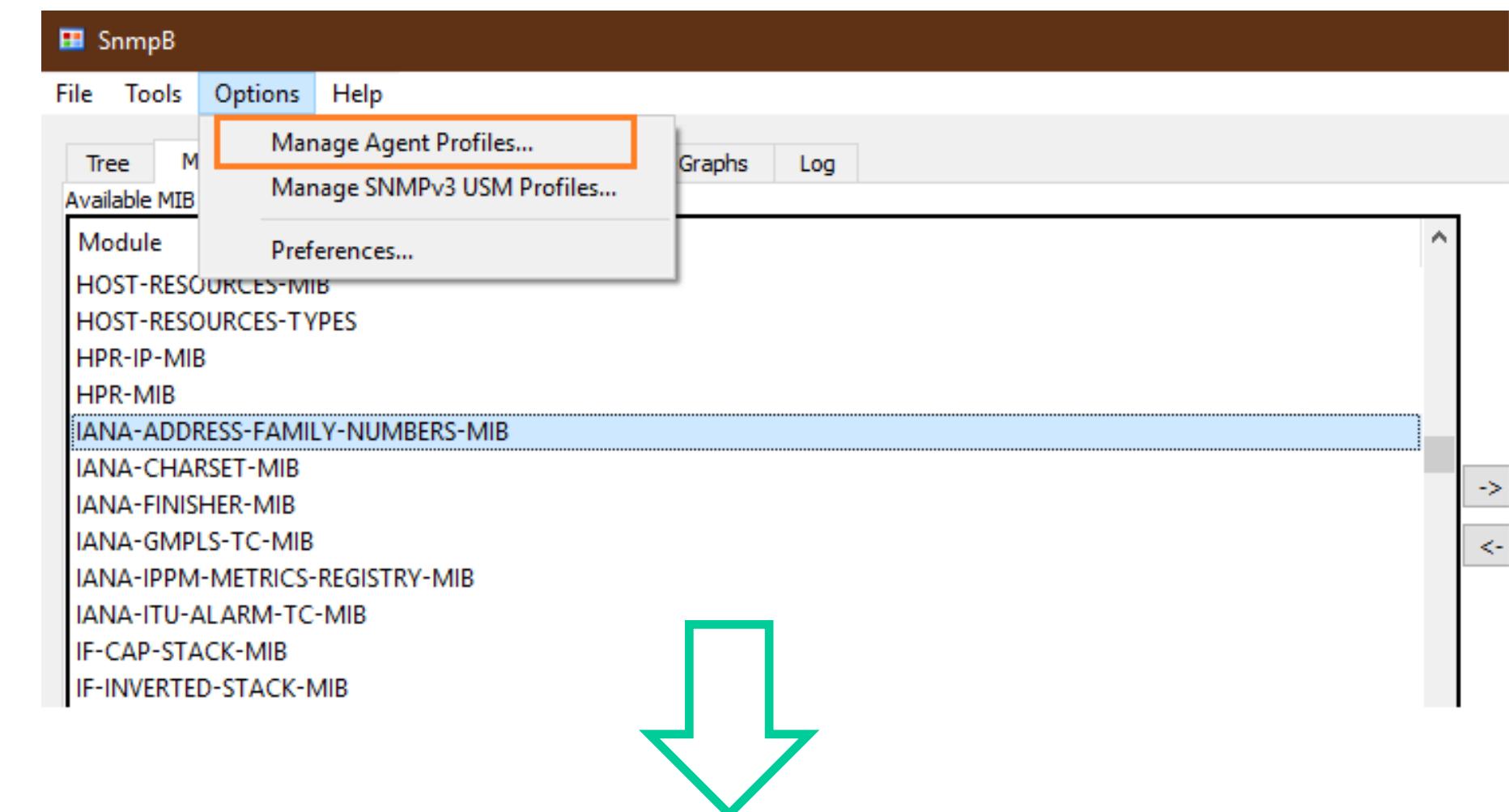
- Go to Option / Manage SNMPv3 USM Profiles
- Do a right click on the blank zone and select “New USM Profile”
- Insert the right values:
  - Set Security User Name to cerberis3.
  - Set Authentication Protocol to SHA.
  - Set Authentication Password to cerberis3.
  - Set Privacy Protocol to AES128.
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



# SNMPB configuration

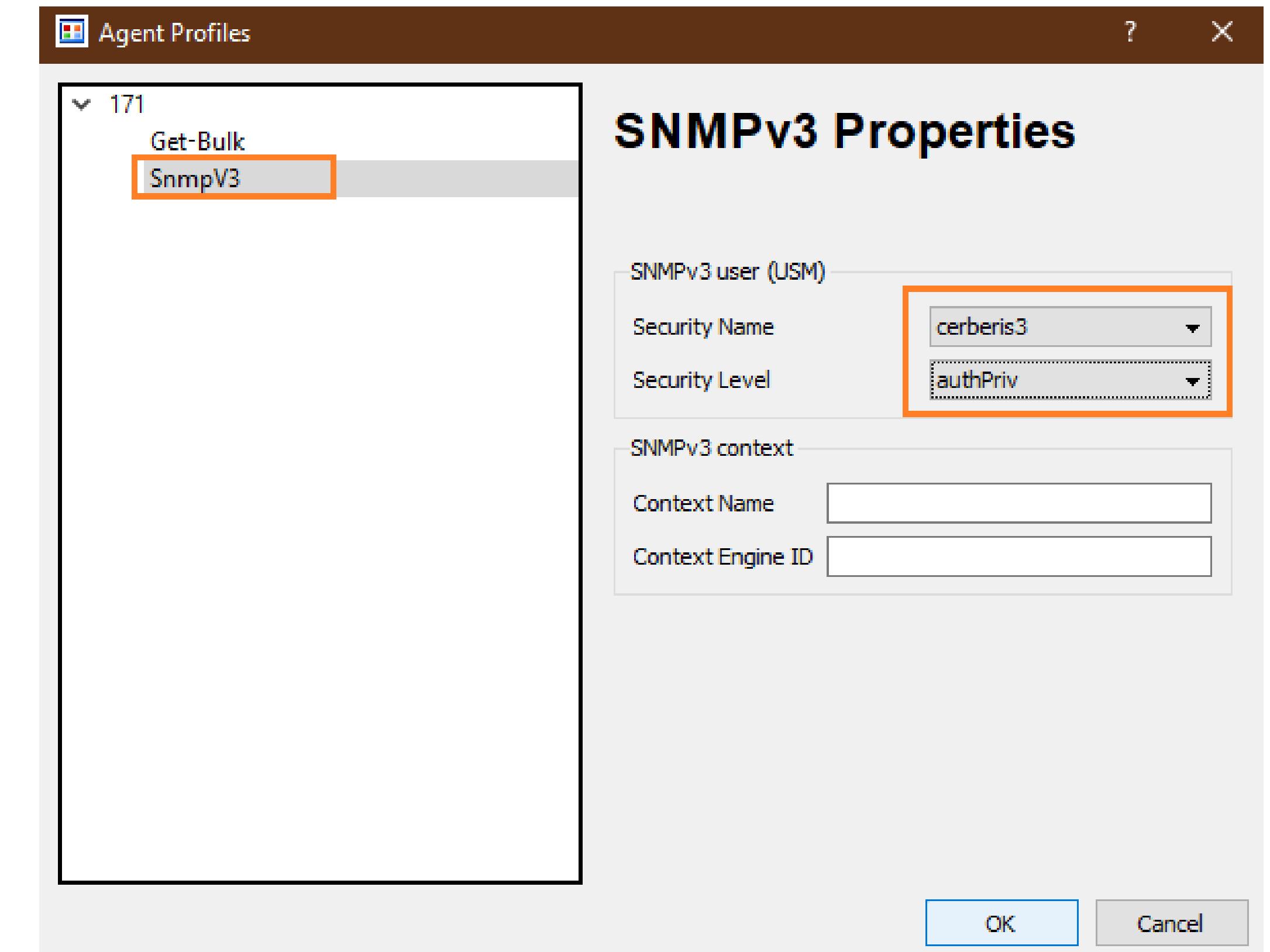
## 5. Create Agent Profile

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



## 6. Configure SNMPv3 on Agent Profile

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



# Configure SNMP Parameters on each CerberisXG/QNC



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

Configure 2x:  
Once for each  
CerberisXG/QNC

Help



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

Usage: snmp [options]

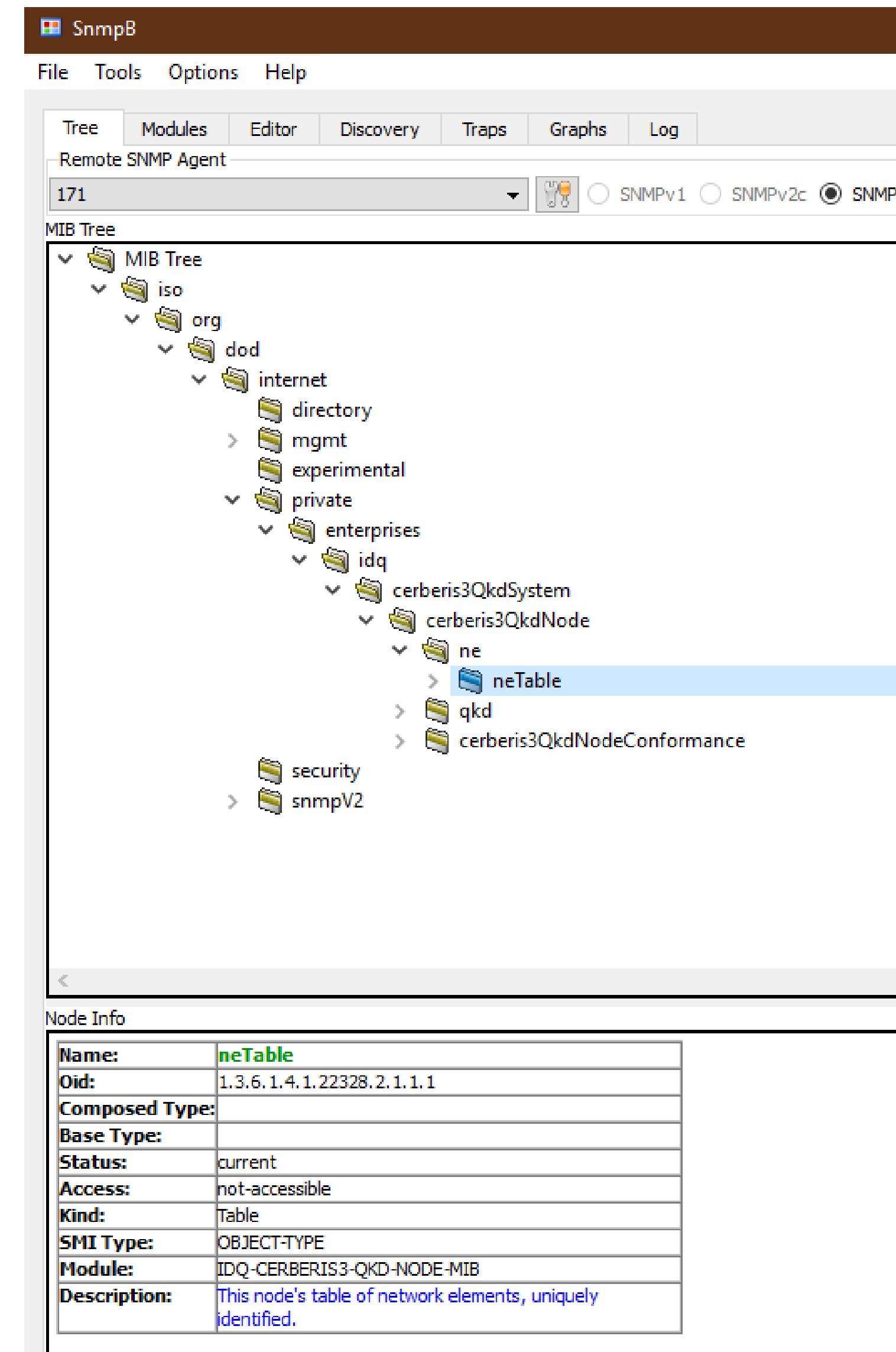
Options:
  --help                                Show help information.
  -c|--contact <arg>                  Set the system contact
  -n|--name <arg>                     Set the system name
  -l|--location <arg>                 Set the system location
  -u|--user <arg>                      Set the user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

admin>
```

# SNMPB configuration

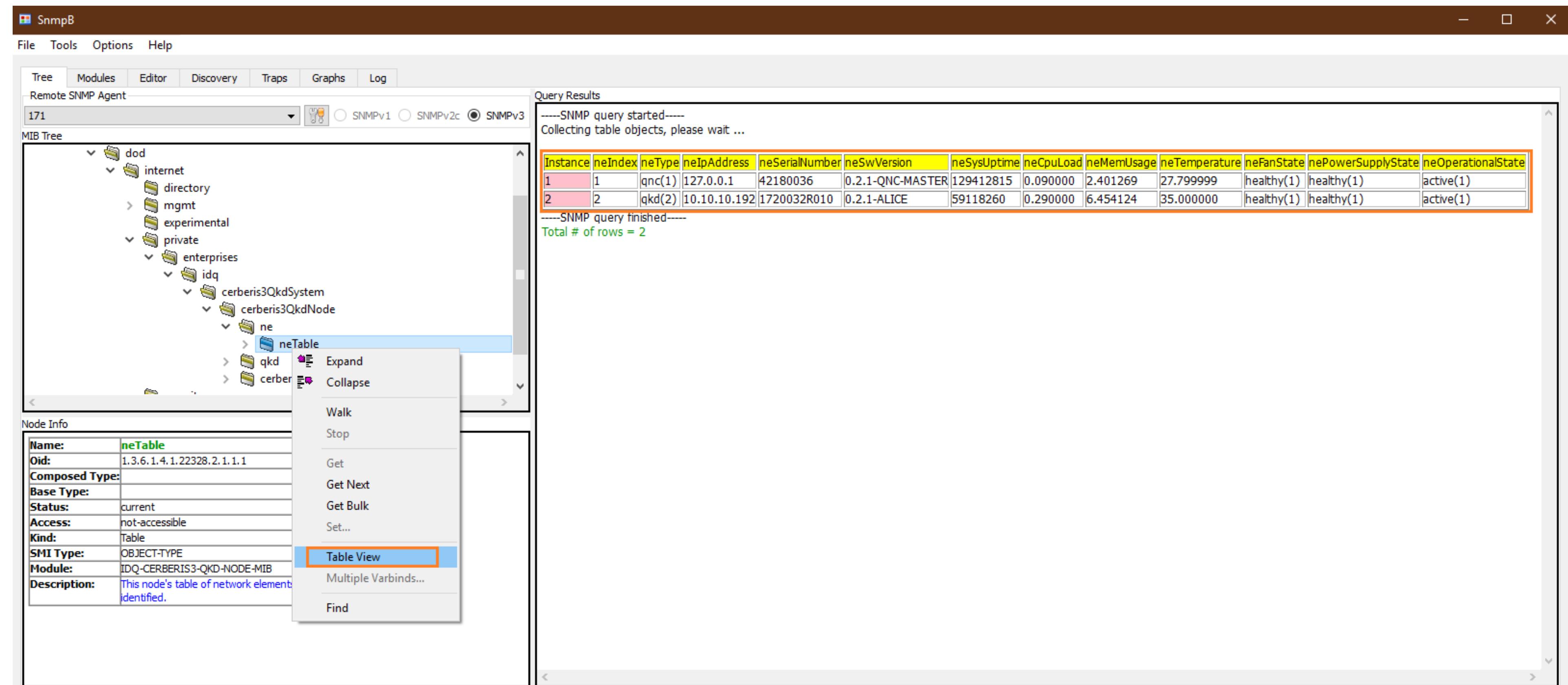
## 7. Check if the communication is working

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



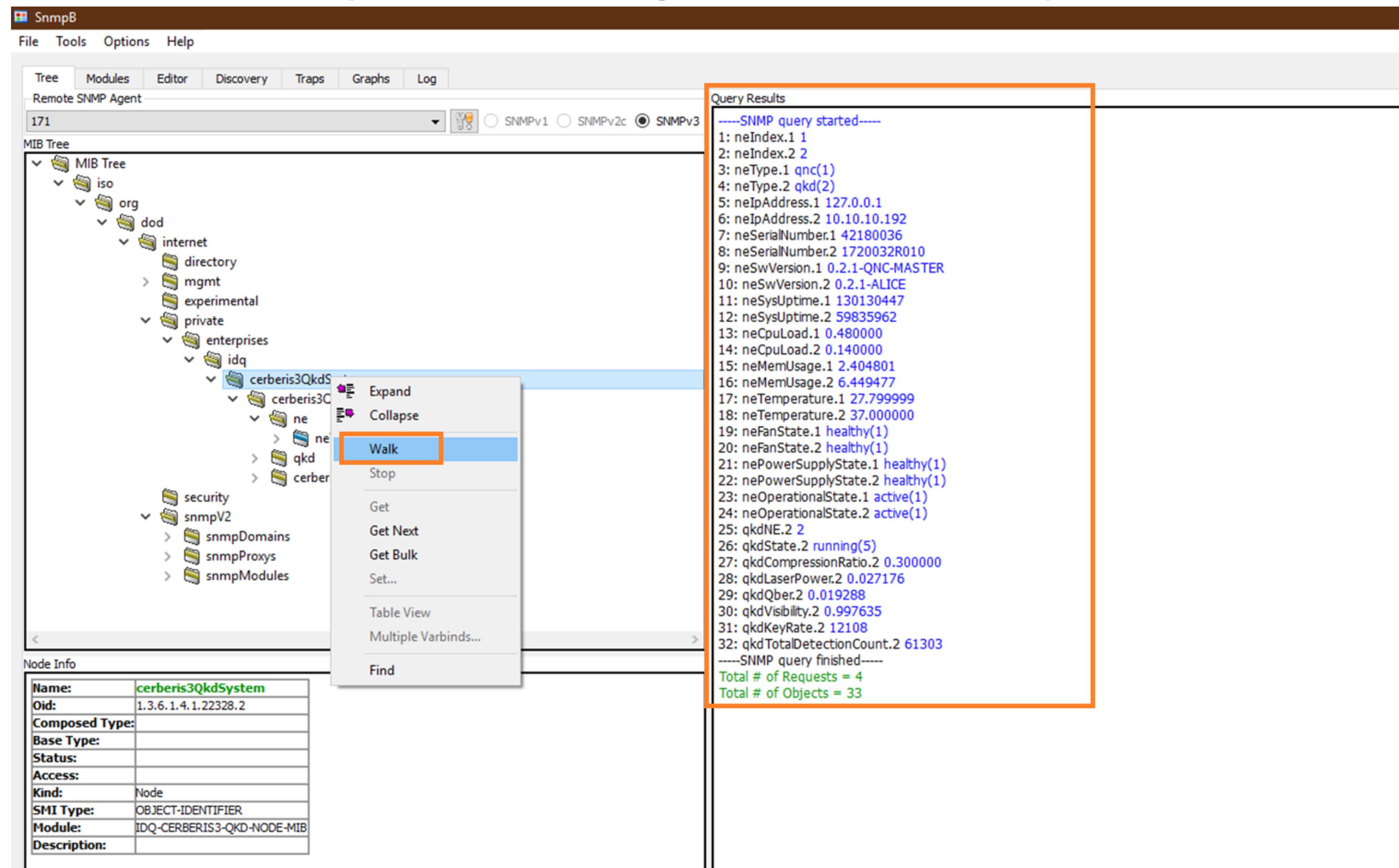
## 7. Check if the communication is working

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



# SNMP Monitoring – Get values

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



# SNMP Monitoring – Get values



- 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 is hosted directly on the CerberisXG**



Cerberis3



CerberisXG

# SNMP Monitoring – Values Explanations

The screenshot shows the SnmpB software interface. On the left, the MIB Tree pane displays a hierarchical tree structure under 'QNC-B' for 'Remote SNMP Agent'. A blue box highlights the 'cerberis3QkdSystem' node under 'cerberis3QkdNode'. On the right, the 'Query Results' pane shows the output of an SNMP query. A green box highlights the first two entries: '1: neIndex.1 1' and '2: neIndex.2 2'. A green arrow points from the 'cerberis3QkdSystem' node in the MIB Tree to the 'neIndex.1 1' entry in the Query Results.

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 query started----  
1: neIndex.1 1  
2: neIndex.2 2  
3: neType.1 qnc(1)  
4: neType.2 qkd(2)  
5: neIpAddress.1 127.0.0.1  
6: neIpAddress.2 10.10.10.197  
7: neSerialNumber.1 42180035  
8: neSerialNumber.2 1720032R110  
9: neSwVersion.1 190205-183011-QNC-SLAVE  
10: neSwVersion.2 0.2.0-BOB  
11: neSysUptime.1 5061569  
12: neSysUptime.2 5060424  
13: neCpuLoad.1 1.160000  
14: neCpuLoad.2 0.470000  
15: neMemUsage.1 1.686915  
16: neMemUsage.2 6.785581  
17: neTemperature.1 27.799999  
18: neTemperature.2 28.000000  
19: neFanState.1 healthy(1)  
20: neFanState.2 healthy(1)  
21: nePowerSupplyState.1 healthy(1)  
22: nePowerSupplyState.2 healthy(1)  
23: neOperationalState.1 active(1)  
24: neOperationalState.2 active(1)  
25: qkdNE.2 2  
26: qkdState.2 running(5)  
27: qkdCompressionRatio.2 0.288146  
28: qkdLaserPower.2 0.026822  
29: qkdQber.2 0.019458  
30: qkdVisibility.2 1.010104  
31: qkdKeyRate.2 6912  
32: qkdTotalDetectionCount.2 56266  
----SNMP query finished----  
Total # of Requests = 4  
Total # of Objects = 33
```

# SNMP Monitoring – Values Explanations

The screenshot shows the SnmpB application window. On the left, the 'MIB Tree' pane displays a hierarchical tree structure under 'QNC-B'. A green box highlights the 'cerberis3QkdSystem' node under 'idq'. To the right, the 'Query Results' pane shows the output of an SNMP query. A green box highlights the first two entries: '3: neType.1 qnc(1)' and '4: neType.2 qkd(2)'. A green arrow points from the 'cerberis3QkdSystem' node in the MIB Tree to the 'neType.1' entry in the Query Results.

3: neType.1  
4: neType.2

🔍 The values indicate the type of the network element:

1: QNC  
2: QKD

Query Results

```
----SNMP query started----  
1: neIndex.1 1  
2: neIndex.2 2  
3: neType.1 qnc(1)  
4: neType.2 qkd(2)  
5: neIpAddress.1 127.0.0.1  
6: neIpAddress.2 10.10.10.197  
7: neSerialNumber.1 42180035  
8: neSerialNumber.2 1720032R110  
9: neSwVersion.1 190205-183011-QNC-SLAVE  
10: neSwVersion.2 0.2.0-BOB  
11: neSysUptime.1 5061569  
12: neSysUptime.2 5060424  
13: neCpuLoad.1 1.160000  
14: neCpuLoad.2 0.470000  
15: neMemUsage.1 1.686915  
16: neMemUsage.2 6.785581  
17: neTemperature.1 27.799999  
18: neTemperature.2 28.000000  
19: neFanState.1 healthy(1)  
20: neFanState.2 healthy(1)  
21: nePowerSupplyState.1 healthy(1)  
22: nePowerSupplyState.2 healthy(1)  
23: neOperationalState.1 active(1)  
24: neOperationalState.2 active(1)  
25: qkdNE.2 2  
26: qkdState.2 running(5)  
27: qkdCompressionRatio.2 0.288146  
28: qkdLaserPower.2 0.026822  
29: qkdQber.2 0.019458  
30: qkdVisibility.2 1.010104  
31: qkdKeyRate.2 6912  
32: qkdTotalDetectionCount.2 56266  
----SNMP query finished----  
Total # of Requests = 4  
Total # of Objects = 33
```

# SNMP Monitoring – Values Explanations

The screenshot shows the SnmpB application interface. On the left, the MIB Tree pane displays a hierarchical structure of MIB objects under 'QNC-B'. A specific node, 'cerberis3QkdSystem', is selected. On the right, the 'Query Results' pane shows the output of an SNMP query. The results list various MIB variables and their values. A green box highlights the first two entries: '5: neIpAddress.1 127.0.0.1' and '6: neIpAddress.2 10.10.10.197'. A green arrow points from this box to the corresponding entries in the MIB Tree pane. Below the highlighted entries, there is explanatory text.

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

Query Results

```
----SNMP query started----  
1: neIndex.1 1  
2: neIndex.2 2  
3: neType.1 qnc(1)  
4: neType.2 qkd(2)  
5: neIpAddress.1 127.0.0.1  
6: neIpAddress.2 10.10.10.197  
7: neSerialNumber.1 42180035  
8: neSerialNumber.2 1720032R110  
9: neSwVersion.1 190205-183011-QNC-SLAVE  
10: neSwVersion.2 0.2.0-BOB  
11: neSysUptime.1 5061569  
12: neSysUptime.2 5060424  
13: neCpuLoad.1 1.160000  
14: neCpuLoad.2 0.470000  
15: neMemUsage.1 1.686915  
16: neMemUsage.2 6.785581  
17: neTemperature.1 27.799999  
neTemperature.2 28.000000  
eFanState.1 healthy(1)  
eFanState.2 healthy(1)  
ePowerSupplyState.1 healthy(1)  
ePowerSupplyState.2 healthy(1)  
eOperationalState.1 active(1)  
eOperationalState.2 active(1)  
kdNE.2 2  
kdState.2 running(5)  
kdCompressionRatio.2 0.288146  
kdLaserPower.2 0.026822  
kdQber.2 0.019458  
kdVisibility.2 1.010104  
kdKeyRate.2 6912  
kdTotalDetectionCount.2 56266  
NMP query finished----  
# of Requests = 4  
# of Objects = 33
```

# SNMP Monitoring – Values Explanations

The screenshot shows the SnmpB application interface. The top menu bar includes File, Tools, Options, Help, Tree, Modules, Editor, Discovery, Traps, Graphs, and Log. The main window has tabs for Remote SNMP Agent (QNC-B) and MIB Tree. The MIB Tree pane displays a hierarchical structure under QNC-B, with a specific node 'cerberis3QkdSystem' highlighted. The Query Results pane shows the output of an SNMP query, listing various MIB objects and their values. A green callout box highlights entries 7 and 8, which are 'neSerialNumber.1' and 'neSerialNumber.2'. Another green callout box contains explanatory text about these serial numbers.

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

These values indicate the network element's serial number.

Base Type:	
Status:	
Access:	
Kind:	Node
SMI Type:	OBJECT-IDENTIFIER
Module:	IDQ-CERBERIS3-QKD-NODE-MIB
Description:	

```
Query Results
----SNMP query started----
1: neIndex.1 1
2: neIndex.2 2
3: neType.1 qnc(1)
4: neType.2 qkd(2)
5: neIpAddress.1 127.0.0.1
6: neIpAddress.2 10.10.10.197
7: neSerialNumber.1 42180035
8: neSerialNumber.2 1720032R110
9: neSwVersion.1 190205-183011-QNC-SLAVE
10: neSwVersion.2 0.2.0-BOB
11: neSysUptime.1 5061569
12: neSysUptime.2 5060424
13: neCpuLoad.1 1.160000
14: neCpuLoad.2 0.470000
15: neMemUsage.1 1.686915
16: neMemUsage.2 6.785581
17: neTemperature.1 27.799999
18: neTemperature.2 28.000000
19: neFanState.1 healthy(1)
20: neFanState.2 healthy(1)
21: nePowerSupplyState.1 healthy(1)
22: nePowerSupplyState.2 healthy(1)
23: neOperationalState.1 active(1)
24: neOperationalState.2 active(1)
25: qkdNE.2 2
26: qkdState.2 running(5)
27: qkdCompressionRatio.2 0.288146
28: qkdLaserPower.2 0.026822
29: qkdQber.2 0.019458
30: qkdVisibility.2 1.010104
31: qkdKeyRate.2 6912
32: qkdTotalDetectionCount.2 56266
----SNMP query finished----
Total # of Requests = 4
Total # of Objects = 33
```

# SNMP Monitoring – Values Explanations

The screenshot shows the SnmpB application window. The top menu bar includes File, Tools, Options, Help, Tree, Modules, Editor, Discovery, Traps, Graphs, and Log. The tabs at the bottom are Tree (selected), Modules, Editor, Discovery, Traps, Graphs, and Log. The main area has tabs for Remote SNMP Agent (QNC-B) and MIB Tree. The MIB Tree pane shows a hierarchical structure under QNC-B, with a blue selection box highlighting the 'cerberis3QkdSystem' node under 'idq'. The 'Query Results' pane displays the output of an SNMP query, with a green box highlighting the entries for neSwVersion.1 and neSwVersion.2. A red arrow points from the highlighted text in the results pane to the corresponding node in the MIB Tree pane.

Query Results

```
----SNMP query started----  
1: neIndex.1 1  
2: neIndex.2 2  
3: neType.1 qnc(1)  
4: neType.2 qkd(2)  
5: neIpAddress.1 127.0.0.1  
6: neIpAddress.2 10.10.10.197  
7: neSerialNumber.1 42180035  
8: neSerialNumber.2 1720032R110  
9: neSwVersion.1 190205-183011-QNC-SLAVE  
10: neSwVersion.2 0.2.0-BOB  
11: neSysUptime.1 5061569  
12: neSysUptime.2 5060424  
13: neCpuLoad.1 1.160000  
14: neCpuLoad.2 0.470000  
15: neMemUsage.1 1.686915  
16: neMemUsage.2 6.785581  
17: neTemperature.1 27.799999  
18: neTemperature.2 28.000000  
19: neFanState.1 healthy(1)  
20: neFanState.2 healthy(1)  
21: nePowerSupplyState.1 healthy(1)  
22: nePowerSupplyState.2 healthy(1)  
23: neOperationalState.1 active(1)  
24: neOperationalState.2 active(1)  
25: qkdNE.2 2  
26: qkdState.2 running(5)  
27: qkdCompressionRatio.2 0.288146  
28: qkdLaserPower.2 0.026822  
29: qkdQber.2 0.019458  
30: qkdVisibility.2 1.010104  
31: qkdKeyRate.2 6912  
32: qkdTotalDetectionCount.2 56266  
----SNMP query finished----  
Total # of Requests = 4  
Total # of Objects = 33
```

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

These value indicate the network element's firmware version

# SNMP Monitoring – Values Explanations

The screenshot shows the SnmpB software interface. The top menu bar includes File, Tools, Options, Help, Tree, Modules, Editor, Discovery, Traps, Graphs, and Log. The main window has tabs for Remote SNMP Agent (QNC-B) and MIB Tree. The MIB Tree pane shows a hierarchy under QNC-B: dod > internet > directory > mgmt > experimental > private > enterprises > idq. The Query Results pane displays the output of an SNMP query:

```
-----SNMP query started-----
1: neIndex.1 1
2: neIndex.2 2
3: neType.1 qnc(1)
4: neType.2 qkd(2)
5: neIpAddress.1 127.0.0.1
6: neIpAddress.2 10.10.10.197
7: neSerialNumber.1 42180035
8: neSerialNumber.2 1720032R110
9: neSwVersion.1 190205-183011-QNC-SLAVE
10: neSwVersion.2 0.2.0-BOB
11: neSysUptime.1 5061569
12: neSysUptime.2 5060424
13: neCpuLoad.1 1.160000
14: neCpuLoad.2 0.470000
15: neMemUsage.1 1.686915
16: neMemUsage.2 6.785581
17: neTemperature.1 27.799999
18: neTemperature.2 28.000000
19: neFanState.1 healthy(1)
20: neFanState.2 healthy(1)
21: nePowerSupplyState.1 healthy(1)
22: nePowerSupplyState.2 healthy(1)
23: neOperationalState.1 active(1)
24: neOperationalState.2 active(1)
25: qkdNE.2 2
26: qkdState.2 running(5)
27: qkdCompressionRatio.2 0.288146
28: qkdLaserPower.2 0.026822
29: qkdQber.2 0.019458
30: qkdVisibility.2 1.010104
31: qkdKeyRate.2 6912
32: qkdTotalDetectionCount.2 56266
-----SNMP query finished-----
Total # of Requests = 4
Total # of Objects = 33
```

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

**Q** 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

The screenshot shows the SnmpB application interface. The top menu bar includes File, Tools, Options, Help, Tree, Modules, Editor, Discovery, Traps, Graphs, and Log. The tabs at the bottom are Tree (selected), Modules, Editor, Discovery, Traps, Graphs, and Log. The status bar indicates Remote SNMP Agent QNC-B, SNMPv1, SNMPv2c, and SNMPv3.

**MIB Tree:** The tree view shows the following structure under QNC-B:  
dod  
  internet  
    directory  
  mgmt  
  experimental  
  private  
    enterprises  
      idq  
      cerberis3QkdSystem  
        cerberis3QkdNode

**Query Results:** The results of an SNMP query are displayed in the right pane. The output is as follows:

```
-----SNMP query started-----
1: neIndex.1 1
2: neIndex.2 2
3: neType.1 qnc(1)
4: neType.2 qkd(2)
5: neIpAddress.1 127.0.0.1
6: neIpAddress.2 10.10.10.197
7: neSerialNumber.1 42180035
8: neSerialNumber.2 1720032R110
9: neSwVersion.1 190205-183011-QNC-SLAVE
10: neSwVersion.2 0.2.0-BOB
11: neSysUptime.1 5061569
12: neSysUptime.2 5060424
13: neCpuLoad.1 1.160000
14: neCpuLoad.2 0.470000
15: neMemUsage.1 1.686915
16: neMemUsage.2 6.785581
17: neTemperature.1 27.799999
18: neTemperature.2 28.000000
19: neFanState.1 healthy(1)
20: neFanState.2 healthy(1)
21: nePowerSupplyState.1 healthy(1)
22: nePowerSupplyState.2 healthy(1)
23: neOperationalState.1 active(1)
24: neOperationalState.2 active(1)
25: qkdNE.2 2
26: qkdState.2 running(5)
27: qkdCompressionRatio.2 0.288146
28: qkdLaserPower.2 0.026822
29: qkdQber.2 0.019458
30: qkdVisibility.2 1.010104
31: qkdKeyRate.2 6912
32: qkdTotalDetectionCount.2 56266
-----SNMP query finished-----
Total # of Requests = 4
Total # of Objects = 33
```

**Annotations:**

- Two specific values are highlighted with a green box and arrow: 13: neCpuLoad.1 for QNC and 14: neCpuLoad.2 for QKD.
- A callout box with a magnifying glass icon provides the definition: "These values indicate the network element's average CPU load over 1 minute."
- Text below the callout box states: "On this ex it's:" followed by two items:
  - 1,16% of CPU usage for QNC
  - 0,47% of CPU usage for QKD

# SNMP Monitoring – Values Explanations

The screenshot shows the SnmpB application window. On the left, the MIB Tree pane displays a hierarchical structure under 'QNC-B' for 'Remote SNMP Agent'. The 'dod' branch is expanded, showing 'internet', 'directory', 'mgmt', 'experimental', 'private', and 'enterprises'. A green box highlights two specific entries: '15: neMemUsage.1' and '16: neMemUsage.2'. An arrow points from this box to the 'Query Results' pane on the right. The 'Query Results' pane shows the output of an SNMP query, starting with '----SNMP query started----' and listing various MIB objects and their values. The highlighted entries correspond to the values shown in the MIB Tree. The pane also ends with '----SNMP query finished----' and summary statistics: 'Total # of Requests = 4' and 'Total # of Objects = 33'.

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

Query Results

```
----SNMP query started----  
1: neIndex.1 1  
2: neIndex.2 2  
3: neType.1 qnc(1)  
4: neType.2 qkd(2)  
5: neIpAddress.1 127.0.0.1  
6: neIpAddress.2 10.10.10.197  
7: neSerialNumber.1 42180035  
8: neSerialNumber.2 1720032R110  
9: neSwVersion.1 190205-183011-QNC-SLAVE  
10: neSwVersion.2 0.2.0-BOB  
11: neSysUptime.1 5061569  
12: neSysUptime.2 5060424  
13: neCpuLoad.1 1.160000  
14: neCpuLoad.2 0.470000  
15: neMemUsage.1 1.686915  
16: neMemUsage.2 6.785581  
17: neTemperature.1 27.799999  
18: neTemperature.2 28.000000  
19: neFanState.1 healthy(1)  
20: neFanState.2 healthy(1)  
21: nePowerSupplyState.1 healthy(1)  
22: nePowerSupplyState.2 healthy(1)  
23: neOperationalState.1 active(1)  
24: neOperationalState.2 active(1)  
25: qkdNE.2 2  
26: qkdState.2 running(5)  
27: qkdCompressionRatio.2 0.288146  
28: qkdLaserPower.2 0.026822  
29: qkdQber.2 0.019458  
30: qkdVisibility.2 1.010104  
31: qkdKeyRate.2 6912  
32: qkdTotalDetectionCount.2 56266  
----SNMP query finished----  
Total # of Requests = 4  
Total # of Objects = 33
```

# SNMP Monitoring – Values Explanations

The screenshot shows the SnmpB application interface. In the top navigation bar, 'Tree' is selected. The main window displays the 'MIB Tree' under 'Remote SNMP Agent' (QNC-B). The tree structure includes 'dod', 'internet', 'mgmt', 'experimental', 'private', 'enterprises', 'idq', and 'ne'. Under 'idq', 'cerberis3QkdSystem' is expanded, showing 'cerberis3QkdNode'. A green box highlights the object identifiers 17 and 18 from the 'Query Results' list.

**Query Results**

```
----SNMP query started----  
1: neIndex.1 1  
2: neIndex.2 2  
3: neType.1 qnc(1)  
4: neType.2 qkd(2)  
5: neIpAddress.1 127.0.0.1  
6: neIpAddress.2 10.10.10.197  
7: neSerialNumber.1 42180035  
8: neSerialNumber.2 1720032R110  
9: neSwVersion.1 190205-183011-QNC-SLAVE  
10: neSwVersion.2 0.2.0-BOB  
11: neSysUptime.1 5061569  
12: neSysUptime.2 5060424  
13: neCpuLoad.1 1.160000  
14: neCpuLoad.2 0.470000  
15: neMemUsage.1 1.686915  
16: neMemUsage.2 6.785581  
17: neTemperature.1 27.799999  
18: neTemperature.2 28.000000  
19: neFanState.1 healthy(1)  
20: neFanState.2 healthy(1)  
21: nePowerSupplyState.1 healthy(1)  
22: nePowerSupplyState.2 healthy(1)  
23: neOperationalState.1 active(1)  
24: neOperationalState.2 active(1)  
25: qkdNE.2 2  
26: qkdState.2 running(5)  
27: qkdCompressionRatio.2 0.288146  
28: qkdLaserPower.2 0.026822  
29: qkdQber.2 0.019458  
30: qkdVisibility.2 1.010104  
31: qkdKeyRate.2 6912  
32: qkdTotalDetectionCount.2 56266  
----SNMP query finished----  
Total # of Requests = 4  
Total # of Objects = 33
```

**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

The screenshot shows the SnmpB software interface. On the left, the MIB Tree pane displays a hierarchical structure of MIB objects under 'QNC-B'. A specific node, 'cerberis3QkdSystem', is selected and highlighted with a green border. On the right, the 'Query Results' pane shows the output of an SNMP query. The results list various MIB variables and their values. Two entries are highlighted with a green border: '19: neFanState.1' and '20: neFanState.2'. An arrow points from the text 'These values indicate the network element's fans state:' to these two entries.

19: neFanState.1  
20: neFanState.2

These values indicate the network element's fans state:

0: unhealthy  
1: healthy

```
----SNMP query started----  
1: neIndex.1 1  
2: neIndex.2 2  
3: neType.1 qnc(1)  
4: neType.2 qkd(2)  
5: neIpAddress.1 127.0.0.1  
6: neIpAddress.2 10.10.10.197  
7: neSerialNumber.1 42180035  
8: neSerialNumber.2 1720032R110  
9: neSwVersion.1 190205-183011-QNC-SLAVE  
10: neSwVersion.2 0.2.0-BOB  
11: neSysUptime.1 5061569  
12: neSysUptime.2 5060424  
13: neCpuLoad.1 1.160000  
14: neCpuLoad.2 0.470000  
15: neMemUsage.1 1.686915  
16: neMemUsage.2 6.785581  
17: neTemperature.1 27.799999  
18: neTemperature.2 28.000000  
19: neFanState.1 healthy(1)  
20: neFanState.2 healthy(1)  
21: nePowerSupplyState.1 healthy(1)  
22: nePowerSupplyState.2 healthy(1)  
23: neOperationalState.1 active(1)  
24: neOperationalState.2 active(1)  
25: qkdNE.2 2  
26: qkdState.2 running(5)  
27: qkdCompressionRatio.2 0.288146  
28: qkdLaserPower.2 0.026822  
29: qkdQber.2 0.019458  
30: qkdVisibility.2 1.010104  
31: qkdKeyRate.2 6912  
32: qkdTotalDetectionCount.2 56266  
----SNMP query finished----  
Total # of Requests = 4  
Total # of Objects = 33
```

# SNMP Monitoring – Values Explanations

The screenshot shows the SnmpB application window. The menu bar includes File, Tools, Options, Help, Tree, Modules, Editor, Discovery, Traps, Graphs, and Log. The toolbar has buttons for Remote SNMP Agent, SNMPv1, SNMPv2c, and SNMPv3. The MIB Tree pane shows a hierarchy under QNC-B: dod > internet > directory > mgmt > experimental > private > enterprises. The Query Results pane displays an SNMP query output:

```
-----SNMP query started-----
1: neIndex.1 1
2: neIndex.2 2
3: neType.1 qnc(1)
4: neType.2 qkd(2)
5: neIpAddress.1 127.0.0.1
6: neIpAddress.2 10.10.10.197
7: neSerialNumber.1 42180035
8: neSerialNumber.2 1720032R110
9: neSwVersion.1 190205-183011-QNC-SLAVE
10: neSwVersion.2 0.2.0-BOB
11: neSysUptime.1 5061569
12: neSysUptime.2 5060424
13: neCpuLoad.1 1.160000
14: neCpuLoad.2 0.470000
15: neMemUsage.1 1.686915
16: neMemUsage.2 6.785581
17: neTemperature.1 27.799999
18: neTemperature.2 28.000000
19: neFanState.1 healthy(1)
20: neFanState.2 healthy(1)
21: nePowerSupplyState.1 healthy(1)
22: nePowerSupplyState.2 healthy(1)
23: neOperationalState.1 active(1)
24: neOperationalState.2 active(1)
25: qkdNE.2 2
26: qkdState.2 running(5)
27: qkdCompressionRatio.2 0.288146
28: qkdLaserPower.2 0.026822
29: qkdQber.2 0.019458
30: qkdVisibility.2 1.010104
31: qkdKeyRate.2 6912
32: qkdTotalDetectionCount.2 56266
-----SNMP query finished-----
Total # of Requests = 4
Total # of Objects = 33
```

Two specific values are highlighted with a green box and labeled:

- 21: nePowerSupplyState.1
- 22: nePowerSupplyState.2

A callout arrow points from the text "This values indicate he network element's power supply state:" to the highlighted values in the query results.

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

- 0: unhealthy
- 1: healthy

# SNMP Monitoring – Values Explanations

The screenshot shows the SnmpB application interface. On the left, the MIB Tree pane displays a hierarchical structure of MIB objects under 'QNC-B'. A specific node, 'cerberis3QkdSystem', is selected and highlighted with a blue selection bar. On the right, the 'Query Results' pane shows the output of an SNMP query. The results list various MIB objects and their values. A green box highlights the entries for 'neOperationalState.1' and 'neOperationalState.2'. A green arrow points from this box to the corresponding entries in the 'Query Results' pane.

23: neOperationalState.1  
24: neOperationalState.2

These values indicate the network element's operational state:

- 0: inactive
- 1: active

```
Query Results
----SNMP query started-----
1: neIndex.1 1
2: neIndex.2 2
3: neType.1 qnc(1)
4: neType.2 qkd(2)
5: neIpAddress.1 127.0.0.1
6: neIpAddress.2 10.10.10.197
7: neSerialNumber.1 42180035
8: neSerialNumber.2 1720032R110
9: neSwVersion.1 190205-183011-QNC-SLAVE
10: neSwVersion.2 0.2.0-BOB
11: neSysUptime.1 5061569
12: neSysUptime.2 5060424
13: neCpuLoad.1 1.160000
14: neCpuLoad.2 0.470000
15: neMemUsage.1 1.686915
16: neMemUsage.2 6.785581
17: neTemperature.1 27.799999
18: neTemperature.2 28.000000
19: neFanState.1 healthy(1)
20: neFanState.2 healthy(1)
21: nePowerSupplyState.1 healthy(1)
22: nePowerSupplyState.2 healthy(1)
23: neOperationalState.1 active(1)
24: neOperationalState.2 active(1)
25: qkdNE.2 2
26: qkdState.2 running(5)
27: qkdCompressionRatio.2 0.288146
28: qkdLaserPower.2 0.026822
29: qkdQber.2 0.019458
30: qkdVisibility.2 1.010104
31: qkdKeyRate.2 6912
32: qkdTotalDetectionCount.2 56266
----SNMP query finished-----
Total # of Requests = 4
Total # of Objects = 33
```

# SNMP Monitoring – Values Explanations

The screenshot shows the SnmpB application interface. On the left, the MIB Tree is displayed with a path selected: `dod>internet>private>enterprises>idq>cerberis3QkdSystem>qkdTable>qkdEntry>qkdNE`. In the center, the **Query Results** pane shows the output of an SNMP query. The results include various network parameters such as `neIndex`, `neType`, `neIpAddress`, and `neOperationalState`. A specific entry, `25: qkdNE.2`, is highlighted with a green border. On the far left, a **Node Info** panel is visible, containing fields like Name, Oid, Compo, Status, Access, Kind, SMI Type, and Module Description.

25: qkdNE.2

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

```
-----SNMP query started-----
1: neIndex.1 1
2: neIndex.2 2
3: neType.1 qnc(1)
4: neType.2 qkd(2)
5: neIpAddress.1 127.0.0.1
6: neIpAddress.2 10.10.10.197
7: neSerialNumber.1 42180035
8: neSerialNumber.2 1720032R110
9: neSwVersion.1 190205-183011-QNC-SLAVE
10: neSwVersion.2 0.2.0-BOB
11: neSysUptime.1 5061569
12: neSysUptime.2 5060424
13: neCpuLoad.1 1.160000
14: neCpuLoad.2 0.470000
15: neMemUsage.1 1.686915
16: neMemUsage.2 6.785581
17: neTemperature.1 27.799999
18: neTemperature.2 28.000000
19: neFanState.1 healthy(1)
20: neFanState.2 healthy(1)
21: nePowerSupplyState.1 healthy(1)
22: nePowerSupplyState.2 healthy(1)
23: neOperationalState.1 active(1)
24: neOperationalState.2 active(1)
25: qkdNE.2 2
26: qkdState.2 running(5)
27: qkdCompressionRatio.2 0.288146
28: qkdLaserPower.2 0.026822
29: qkdQber.2 0.019458
30: qkdVisibility.2 1.010104
31: qkdKeyRate.2 6912
32: qkdTotalDetectionCount.2 56266
-----SNMP query finished-----
Total # of Requests = 4
Total # of Objects = 33
```

# SNMP Monitoring – Values Explanations

The screenshot shows the SnmpB application interface. On the left, the MIB Tree pane displays a hierarchical structure under 'QNC-B' with nodes like 'dod' and 'internet'. A green box highlights the entry '26: qkdState.2'. To the right, the 'Query Results' pane shows the output of an SNMP query. An arrow points from the highlighted entry in the tree to its corresponding value in the results pane.

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

Query Results

```
----SNMP query started----  
1: neIndex.1 1  
2: neIndex.2 2  
3: neType.1 qnc(1)  
4: neType.2 qkd(2)  
5: neIpAddress.1 127.0.0.1  
6: neIpAddress.2 10.10.10.197  
7: neSerialNumber.1 42180035  
8: neSerialNumber.2 1720032R110  
9: neSwVersion.1 190205-183011-QNC-SLAVE  
10: neSwVersion.2 0.2.0-BOB  
11: neSysUptime.1 5061569  
12: neSysUptime.2 5060424  
13: neCpuLoad.1 1.160000  
14: neCpuLoad.2 0.470000  
15: neMemUsage.1 1.686915  
16: neMemUsage.2 6.785581  
17: neTemperature.1 27.799999  
18: neTemperature.2 28.000000  
19: neFanState.1 healthy(1)  
20: neFanState.2 healthy(1)  
21: nePowerSupplyState.1 healthy(1)  
22: nePowerSupplyState.2 healthy(1)  
23: neOperationalState.1 active(1)  
24: neOperationalState.2 active(1)  
25: qkdNE.2 2  
26: qkdState.2 running(5) running(5)  
27: qkdCompressionRatio.2 0.288146  
28: qkdLaserPower.2 0.026822  
29: qkdQber.2 0.019458  
30: qkdVisibility.2 1.010104  
31: qkdKeyRate.2 6912  
32: qkdTotalDetectionCount.2 56266  
----SNMP query finished----  
Total # of Requests = 4  
Total # of Objects = 33
```

# 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

The screenshot shows the SnmpB application interface. On the left, the MIB Tree pane displays a hierarchical structure under 'QNC-B' for 'Remote SNMP Agent'. The 'dod' and 'internet' branches are expanded, showing 'directory', 'mgmt', 'experimental', and 'private' subfolders. On the right, the 'Query Results' pane shows the output of an SNMP query:

```
-----SNMP query started-----
1: neIndex.1 1
2: neIndex.2 2
3: neType.1 qnc(1)
4: neType.2 qkd(2)
5: neIpAddress.1 127.0.0.1
6: neIpAddress.2 10.10.10.197
7: neSerialNumber.1 42180035
8: neSerialNumber.2 1720032R110
9: neSwVersion.1 190205-183011-QNC-SLAVE
10: neSwVersion.2 0.2.0-BOB
11: neSysUptime.1 5061569
12: neSysUptime.2 5060424
13: neCpuLoad.1 1.160000
14: neCpuLoad.2 0.470000
15: neMemUsage.1 1.686915
16: neMemUsage.2 6.785581
17: neTemperature.1 27.799999
18: neTemperature.2 28.000000
19: neFanState.1 healthy(1)
20: neFanState.2 healthy(1)
21: nePowerSupplyState.1 healthy(1)
22: nePowerSupplyState.2 healthy(1)
23: neOperationalState.1 active(1)
24: neOperationalState.2 active(1)
25: qkdNE.2 2
26: qkdState.2 running(5)
27: qkdCompressionRatio.2 0.288146
28: qkdLaserPower.2 0.026822
29: qkdQber.2 0.019458
30: qkdVisibility.2 1.010104
31: qkdKeyRate.2 6912
32: qkdTotalDetectionCount.2 56266
-----SNMP query finished-----
Total # of Requests = 4
Total # of Objects = 33
```

A green box highlights the line '27: qkdCompressionRatio.2 0.288146'. A green arrow points from this highlighted line to the corresponding entry in the 'Query Results' pane.

**27: qkdCompressionRatio.2**  
🔍 This value 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 ensures that no vulnerable key will be exchanged and used,

# SNMP Monitoring – Values Explanations

The screenshot shows the SnmpB application interface. On the left, the MIB Tree is displayed with a path selected: **dod > internet > directory > mgmt > experimental > private > enterprises > idq > cerberis3QkdSystem > cerberis3QkdNode > ne > qkd > qkdTable > qkdEntry > qkdNE > qkdState > qkdCompressionRatio**. In the center, the **Query Results** pane shows the output of an SNMP query:

```
-----SNMP query started-----
1: neIndex.1 1
2: neIndex.2 2
3: neType.1 qnc(1)
4: neType.2 qkd(2)
5: neIpAddress.1 127.0.0.1
6: neIpAddress.2 10.10.10.197
7: neSerialNumber.1 42180035
8: neSerialNumber.2 1720032R110
9: neSwVersion.1 190205-183011-QNC-SLAVE
10: neSwVersion.2 0.2.0-BOB
11: neSysUptime.1 5061569
12: neSysUptime.2 5060424
13: neCpuLoad.1 1.160000
14: neCpuLoad.2 0.470000
15: neMemUsage.1 1.686915
16: neMemUsage.2 6.785581
17: neTemperature.1 27.799999
18: neTemperature.2 28.000000
19: neFanState.1 healthy(1)
20: neFanState.2 healthy(1)
21: nePowerSupplyState.1 healthy(1)
22: nePowerSupplyState.2 healthy(1)
23: neOperationalState.1 active(1)
24: neOperationalState.2 active(1)
25: qkdNE.2 2
26: qkdState.2 running(5)
27: qkdCompressionRatio.2 0.288146
28: qkdLaserPower.2 0.026822
29: qkdQber.2 0.019458
30: qkdVisibility.2 1.010104
31: qkdKeyRate.2 6912
32: qkdTotalDetectionCount.2 56266
-----SNMP query finished-----
Total # of Requests = 4
Total # of Objects = 33
```

A green box highlights the entry **28: qkdLaserPower.2**, and a green magnifying glass icon points to the explanatory text below it.

**28: qkdLaserPower.2**  
This value indicates the QKD system's laser power.

# SNMP Monitoring – Values Explanations

The screenshot shows the SnmpB application interface. On the left, the MIB Tree is displayed with a path selected: `dod>internet>directory>mgmt>experimental>private>enterprises>idq>cerberis3QkdSystem>cerberis3QkdNode>ne>qkd`. A green box highlights the node `cerberis3QkdSystem`. On the right, the `Query Results` pane shows the output of an SNMP query. The results include various system parameters and their values. A green box highlights the value `0.019458` for the object `qkdQber.2`.

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

Name:	cerberis3QkdSystem
Oid:	1.3.6.1.4.1.8072.2.1.1.1.1.1
Composed Type:	
Base Type:	
Status:	
Access:	
Kind:	Node
SMI Type:	OBJECT
Module:	IDQ-CERBERIS-QKD
Description:	

Query Results

```
-----SNMP query started-----
1: neIndex.1 1
2: neIndex.2 2
3: neType.1 qnc(1)
4: neType.2 qkd(2)
5: neIpAddress.1 127.0.0.1
6: neIpAddress.2 10.10.10.197
7: neSerialNumber.1 42180035
8: neSerialNumber.2 1720032R110
9: neSwVersion.1 190205-183011-QNC-SLAVE
10: neSwVersion.2 0.2.0-BOB
11: neSysUptime.1 5061569
12: neSysUptime.2 5060424
13: neCpuLoad.1 1.160000
14: neCpuLoad.2 0.470000
15: neMemUsage.1 1.686915
16: neMemUsage.2 6.785581
17: neTemperature.1 27.799999
18: neTemperature.2 28.000000
19: neFanState.1 healthy(1)
20: neFanState.2 healthy(1)
21: nePowerSupplyState.1 healthy(1)
22: nePowerSupplyState.2 healthy(1)
23: neOperationalState.1 active(1)
24: neOperationalState.2 active(1)
25: qkdNE.2 2
26: qkdState.2 running(5)
27: qkdCompressionRatio.2 0.288146
28: qkdLaserPower.2 0.026822
29: qkdQber.2 0.019458
30: qkdVisibility.2 1.010104
31: qkdKeyRate.2 6912
32: qkdTotalDetectionCount.2 56266
-----SNMP query finished-----
Total # of Requests = 4
Total # of Objects = 33
```

# SNMP Monitoring – Values Explanations

The screenshot shows the SnmpB software interface. The top menu bar includes File, Tools, Options, Help, Tree, Modules, Editor, Discovery, Traps, Graphs, and Log. The tabs at the bottom are Remote SNMP Agent (selected), SNMPv1, SNMPv2c, and SNMPv3. The main window has a 'MIB Tree' tab selected, displaying a hierarchical tree structure under 'QNC-B'. A specific node, 'cerberis3QkdSystem', is highlighted. To the right, the 'Query Results' pane displays the output of an SNMP query. The results list various MIB objects and their values, with object 30 highlighted in green.

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

Query Results

```
-----SNMP query started-----
1: neIndex.1 1
2: neIndex.2 2
3: neType.1 qnc(1)
4: neType.2 qkd(2)
5: neIpAddress.1 127.0.0.1
6: neIpAddress.2 10.10.10.197
7: neSerialNumber.1 42180035
8: neSerialNumber.2 1720032R110
9: neSwVersion.1 190205-183011-QNC-SLAVE
10: neSwVersion.2 0.2.0-BOB
11: neSysUptime.1 5061569
12: neSysUptime.2 5060424
13: neCpuLoad.1 1.160000
14: neCpuLoad.2 0.470000
15: neMemUsage.1 1.686915
16: neMemUsage.2 6.785581
17: neTemperature.1 27.799999
18: neTemperature.2 28.000000
19: neFanState.1 healthy(1)
20: neFanState.2 healthy(1)
21: nePowerSupplyState.1 healthy(1)
22: nePowerSupplyState.2 healthy(1)
23: neOperationalState.1 active(1)
24: neOperationalState.2 active(1)
25: qkdNE.2 2
26: qkdState.2 running(5)
27: qkdCompressionRatio.2 0.288146
28: qkdLaserPower.2 0.026822
29: qkdOber.2 0.019458
30: qkdVisibility.2 1.010104
31: qkdKeyRate.2 6912
32: qkdTotalDetectionCount.2 56266
-----SNMP query finished-----
Total # of Requests = 4
Total # of Objects = 33
```

# SNMP Monitoring – Values Explanations

The screenshot shows the SnmpB software interface. The top menu bar includes File, Tools, Options, Help, Tree, Modules, Editor, Discovery, Traps, Graphs, and Log. The tabs at the top are Remote SNMP Agent (selected), QNC-B, SNMPv1, SNMPv2c, and SNMPv3. The MIB Tree panel shows a hierarchical structure under QNC-B, with the 'cerberis3QkdSystem' node selected. The Node Info panel displays details for this node, including Name: cerberis3QkdSystem, Oid: 1.3.6.1.4.1.22228.2, Composed Type:, Base Type:, Status:, Access:, Kind: Node, SMI Type: OBJECT, Module: IDQ-CERBERIS, and Description:. A callout box highlights the 'Status:' field. The Query Results panel shows the output of an SNMP query, listing various MIB objects and their values. The value for qkdKeyRate.2 is highlighted with a green box and an arrow pointing to it from the Node Info status field.

31: qkdKeyRate.2

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

About 8 bits per second is a minimum

Query Results

```
-----SNMP query started-----
1: neIndex.1 1
2: neIndex.2 2
3: neType.1 qnc(1)
4: neType.2 qkd(2)
5: neIpAddress.1 127.0.0.1
6: neIpAddress.2 10.10.10.197
7: neSerialNumber.1 42180035
8: neSerialNumber.2 1720032R110
9: neSwVersion.1 190205-183011-QNC-SLAVE
10: neSwVersion.2 0.2.0-BOB
11: neSysUptime.1 5061569
12: neSysUptime.2 5060424
13: neCpuLoad.1 1.160000
14: neCpuLoad.2 0.470000
15: neMemUsage.1 1.686915
16: neMemUsage.2 6.785581
17: neTemperature.1 27.799999
18: neTemperature.2 28.000000
19: neFanState.1 healthy(1)
20: neFanState.2 healthy(1)
21: nePowerSupplyState.1 healthy(1)
22: nePowerSupplyState.2 healthy(1)
23: neOperationalState.1 active(1)
24: neOperationalState.2 active(1)
25: qkdNE.2 2
26: qkdState.2 running(5)
27: qkdCompressionRatio.2 0.288146
28: qkdLaserPower.2 0.026822
29: qkdQber.2 0.019458
30: qkdVisibility.2 1.010104
31: qkdKeyRate.2 6912
32: qkdTotalDetectionCount.2 56266
-----SNMP query finished-----
Total # of Requests = 4
Total # of Objects = 33
```

# SNMP Monitoring – Values Explanations

The screenshot shows the SnmpB software interface. The top menu bar includes File, Tools, Options, Help, Tree, Modules, Editor, Discovery, Traps, Graphs, and Log. The toolbar below has icons for Remote SNMP Agent, SNMPv1, SNMPv2c, and SNMPv3. The MIB Tree pane displays a hierarchical structure under QNC-B, specifically the cerberis3QkdSystem subtree. The Node Info pane shows details for a selected node, including Name: 32: qkdTotalDetectionCount.2, Oid: 32: qkdTotalDetectionCount.2, Composed Type: qkdTotalDetectionCount, Base Type: qkdTotalDetectionCount, Status: healthy(1), Access: read, Kind: scalar, SMI Type: qkdTotalDetectionCount, Module: cerberis3QkdSystem, and Description: This value indicates the QKD system's detection count on DATA and MONITOR. A green callout box highlights this description. The Query Results pane lists the results of an SNMP query, starting with neIndex and ending with qkdTotalDetectionCount.2.

Query Results

```
-----SNMP query started-----
1: neIndex.1 1
2: neIndex.2 2
3: neType.1 qnc(1)
4: neType.2 qkd(2)
5: neIpAddress.1 127.0.0.1
6: neIpAddress.2 10.10.10.197
7: neSerialNumber.1 42180035
8: neSerialNumber.2 1720032R110
9: neSwVersion.1 190205-183011-QNC-SLAVE
10: neSwVersion.2 0.2.0-BOB
11: neSysUptime.1 5061569
12: neSysUptime.2 5060424
13: neCpuLoad.1 1.160000
14: neCpuLoad.2 0.470000
15: neMemUsage.1 1.686915
16: neMemUsage.2 6.785581
17: neTemperature.1 27.799999
18: neTemperature.2 28.000000
19: neFanState.1 healthy(1)
20: neFanState.2 healthy(1)
21: nePowerSupplyState.1 healthy(1)
22: nePowerSupplyState.2 healthy(1)
23: neOperationalState.1 active(1)
24: neOperationalState.2 active(1)
25: qkdNE.2 2
26: qkdState.2 running(5)
27: qkdCompressionRatio.2 0.288146
28: qkdLaserPower.2 0.026822
29: qkdQber.2 0.019458
30: qkdVisibility.2 1.010104
31: qkdKeyRate.2 6912
32: qkdTotalDetectionCount.2 56266
-----SNMP query finished-----
Total # of Requests = 4
Total # of Objects = 33
```



- 2 values **are important to check:**
  - **Visibility** (it has to be close with 1)
  - **QBER** (approx 2-3 %) (4.5% max)
  - **Key Rate** (minimum 8 bits per second)
- If these two values are correct the QKDs **should be able to deliver keys to encryptors**
- Key delivery should be checked on the encryptors

ETSI client



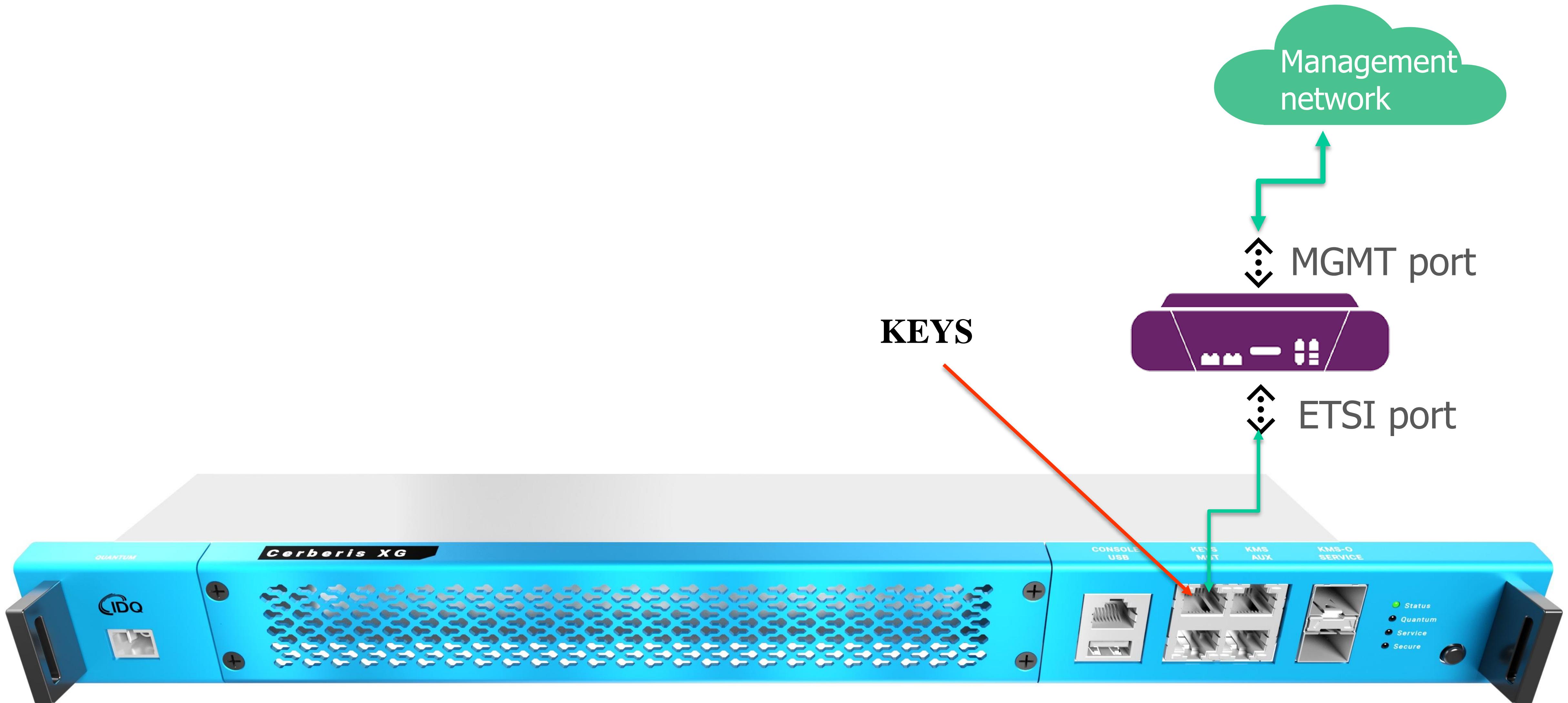
- **ETSI** is the primary protocol to connect third party devices to the CerberisXG or QNCs (We support others such as Cisco, etc)
- Usually the third party devices are:
  - **Encryptors**
  - **HSM**
  - Encryptor **simulators**
- **Regardless** of the type of third party devices, the processes are quite similar
- **ETSI** will provide 256 bits keys based on Quantum Physics to third party devices

You have **2 choices**:

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

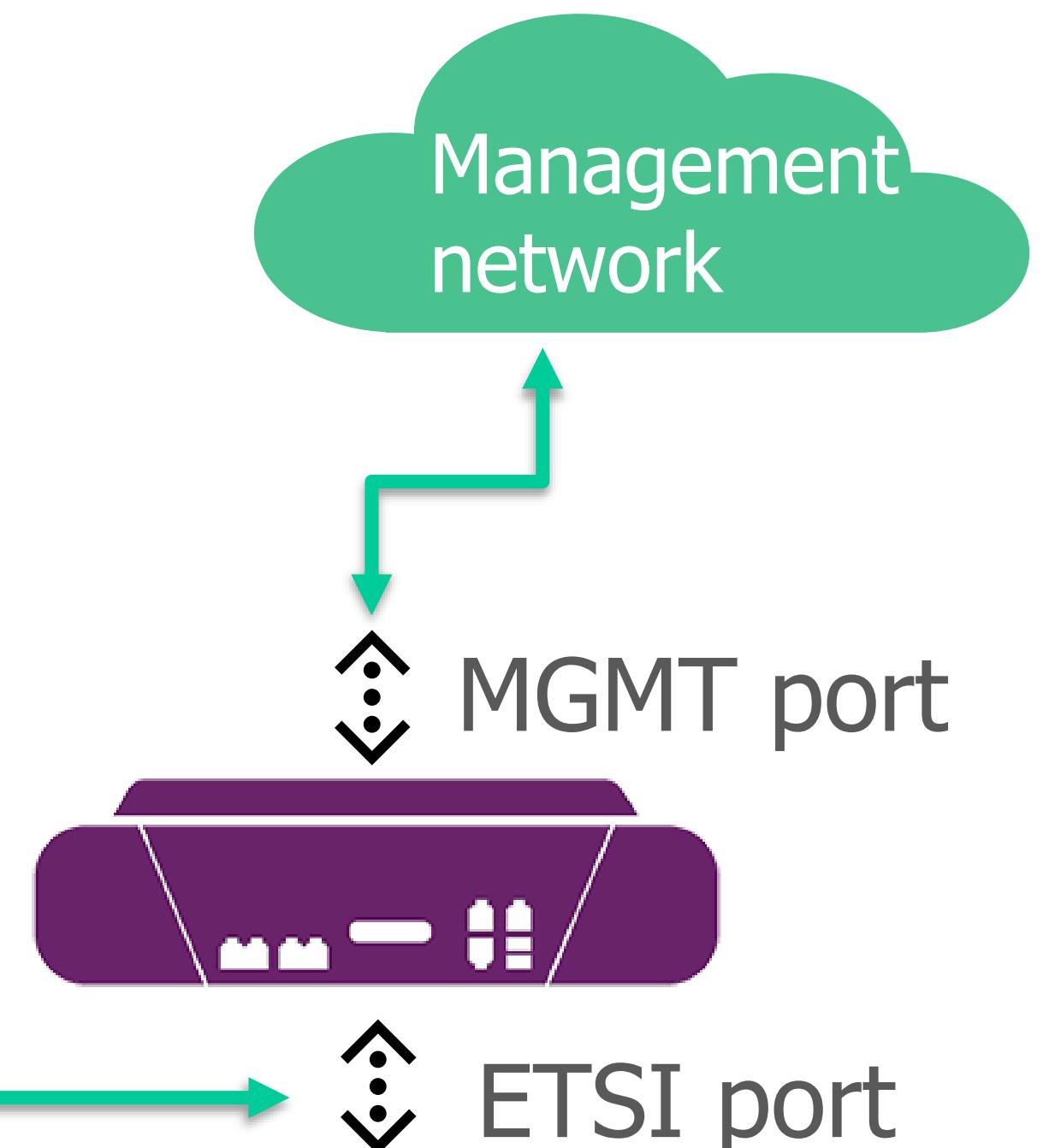
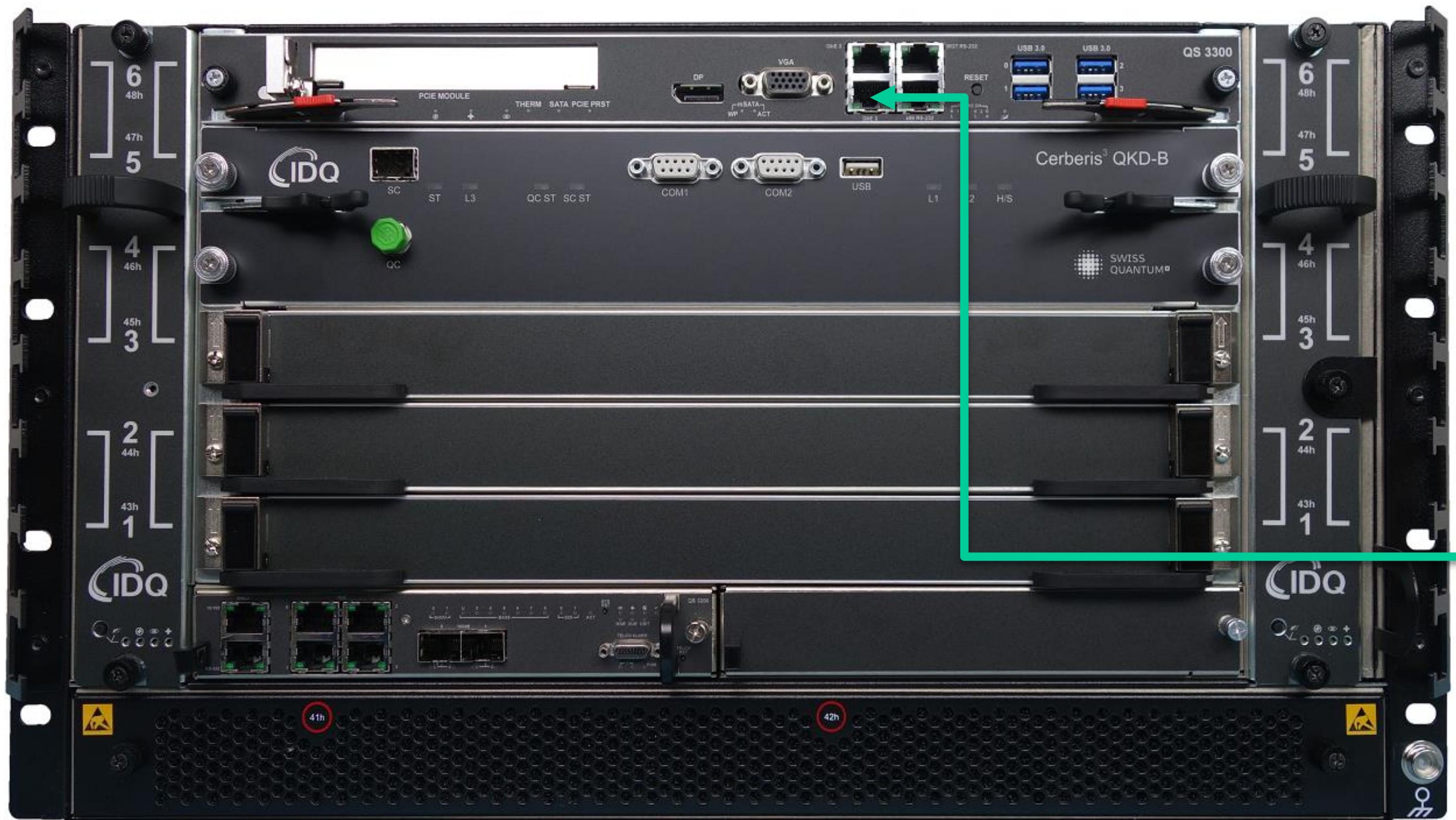
You have 2 choices:

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



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

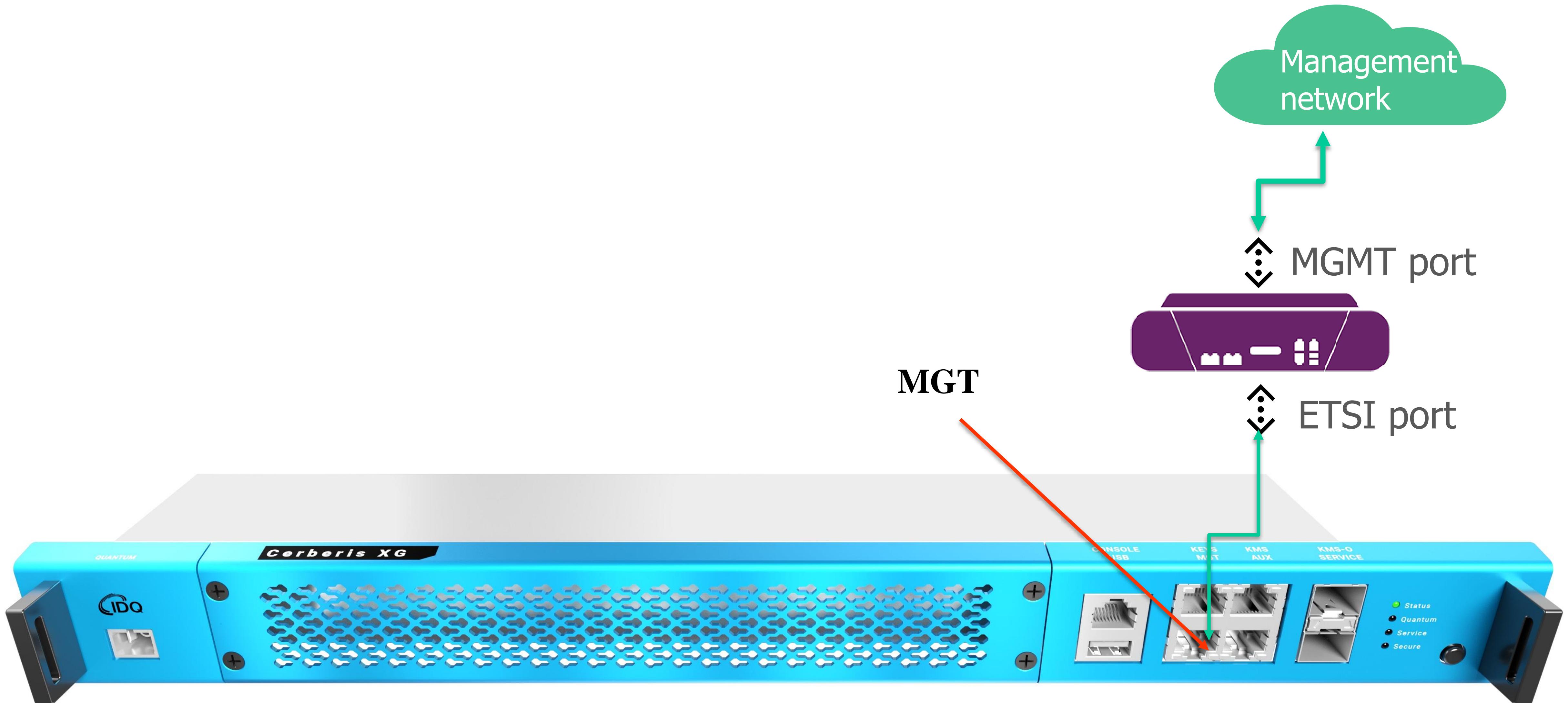
→ Connect directly frontpane port to ETSI client port (encryptor)



DO IT FOR EACH QNC

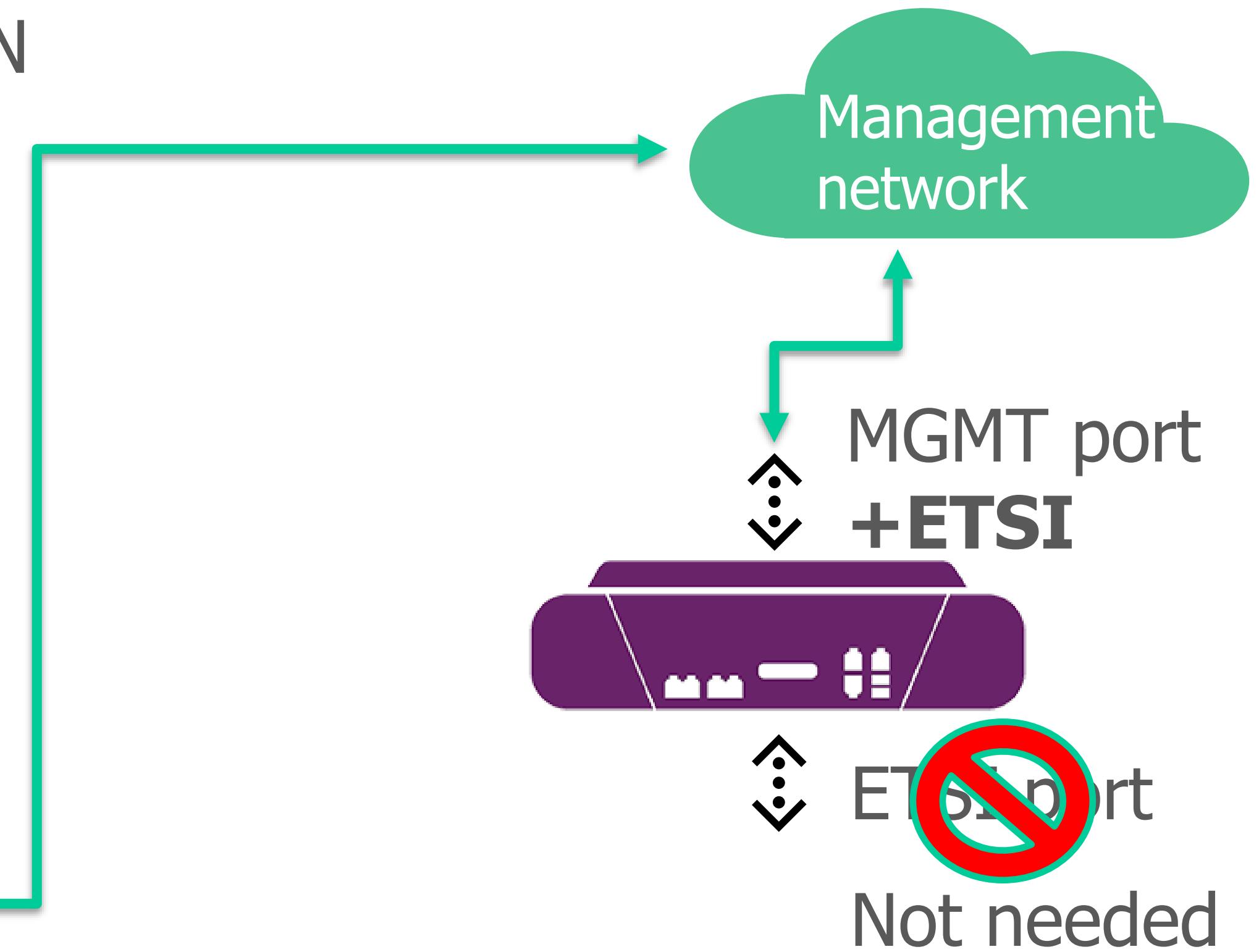
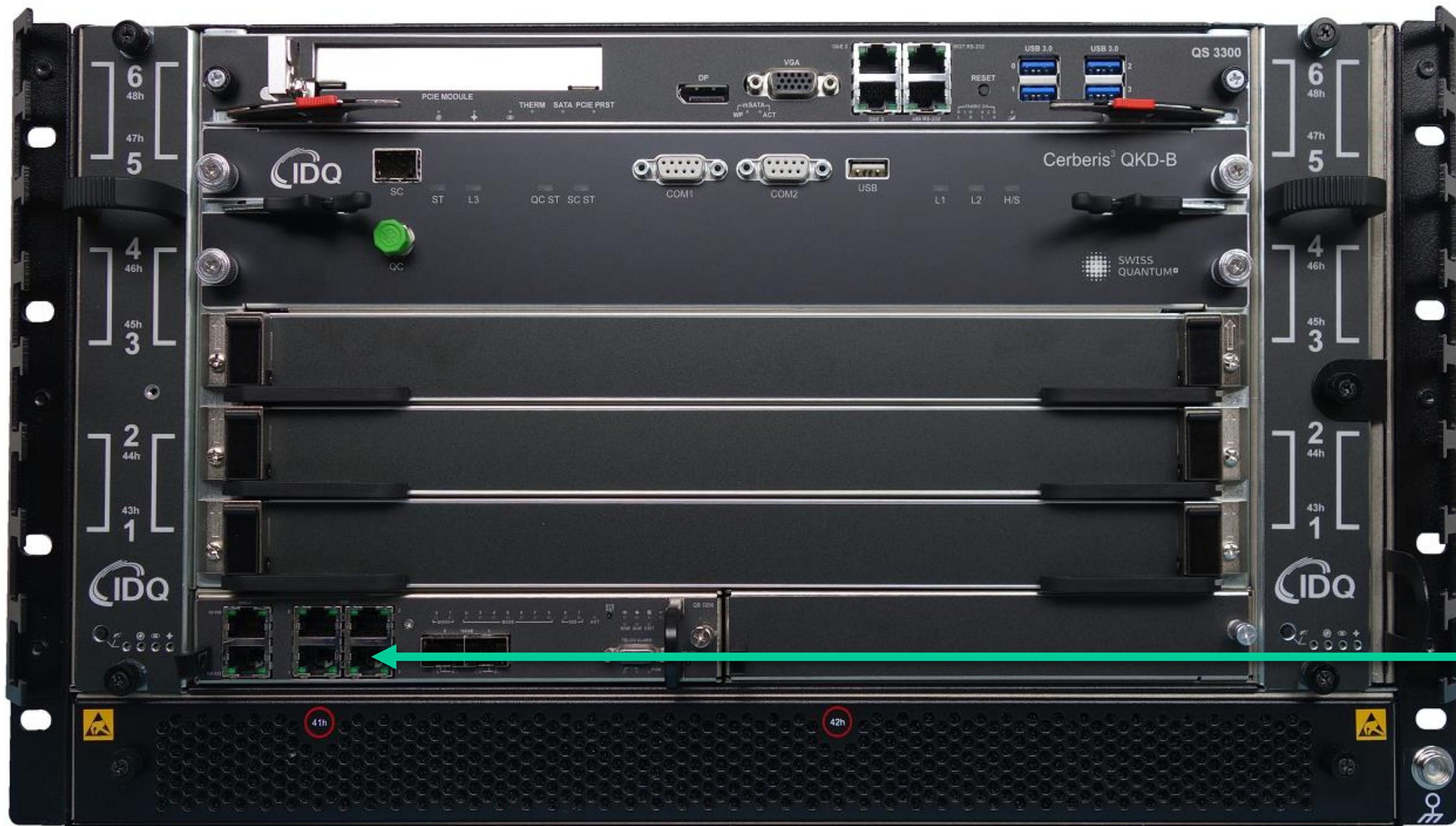
You have 2 choices:

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



**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**

## 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**

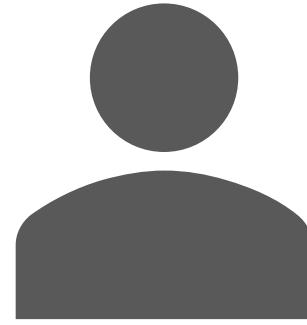
**Recommend 1 CA file covering both Client and QKD certs/keys**

Files requirement:

Alternative “Optional for **test** only”

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

## Understand the ETSI TLS process



**Client**



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



**IDQ**

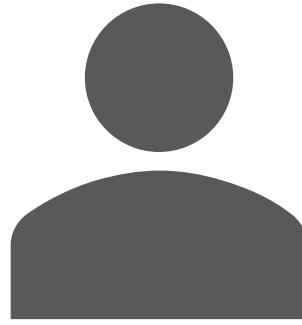


IDQ **Certification Authority**  
IDQ CA file (cert file)  
ie: IDQCA.cert.pem

**Each of the parties must have and operate his own CA**

**They are independent, nobody knows each other**

## Understand the ETSI TLS process



**Client**



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

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

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

These files will be imported into the customer's  
encryptor



**IDQ**



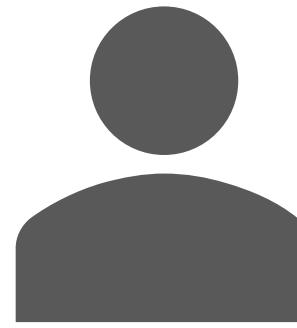
IDQ Certification Authority  
IDQ CA file (cert file)  
IDQCA.cert.pem

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

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

These files will be already hosted into the QNC

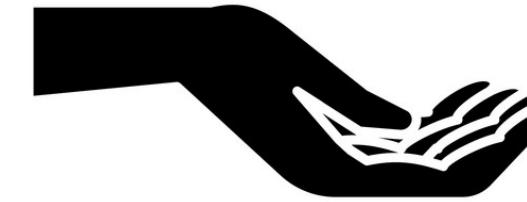
## Understand the ETSI TLS process



**Client**

own:

**ClientCA.cert.pem**  
ClientETSI.key.pem  
ClientETSI.cert.pem



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



**IDQ**

own:

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

**Encryptor**

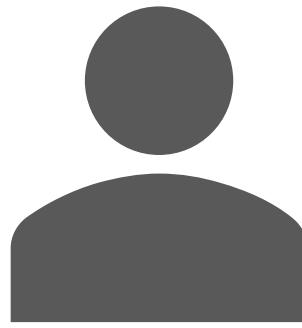
ClientETSI.key.pem  
ClientETSI.cert.pem

**QNC**

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

**ClientCA.cert.pem**

## 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**

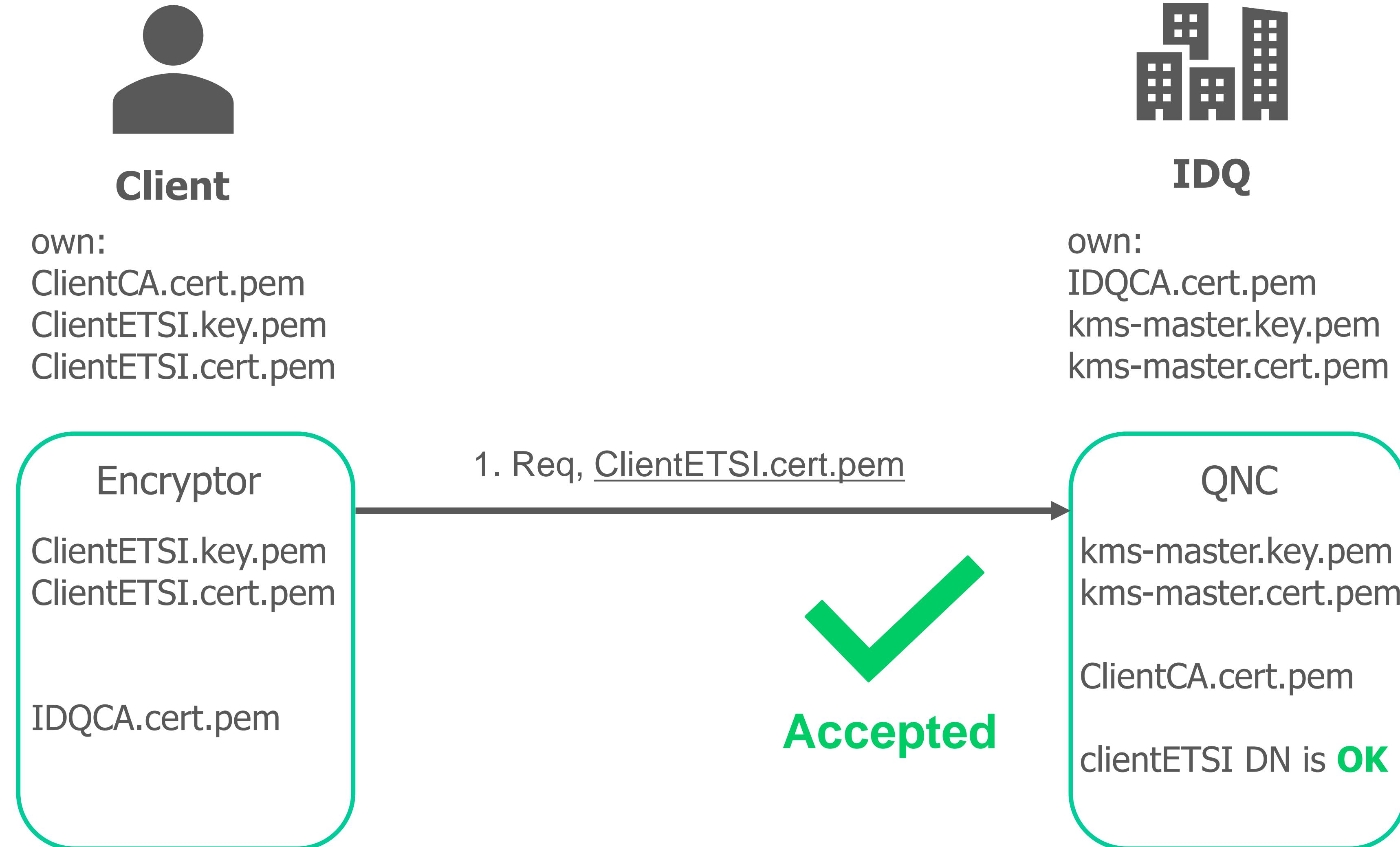
**QNC**

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

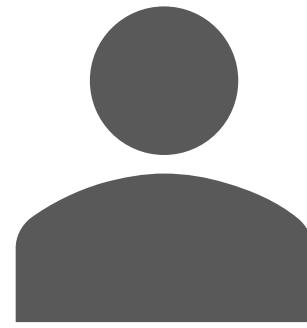
**ClientCA.cert.pem**



## Understand the ETSI TLS process



## Understand the ETSI TLS process



**Client**

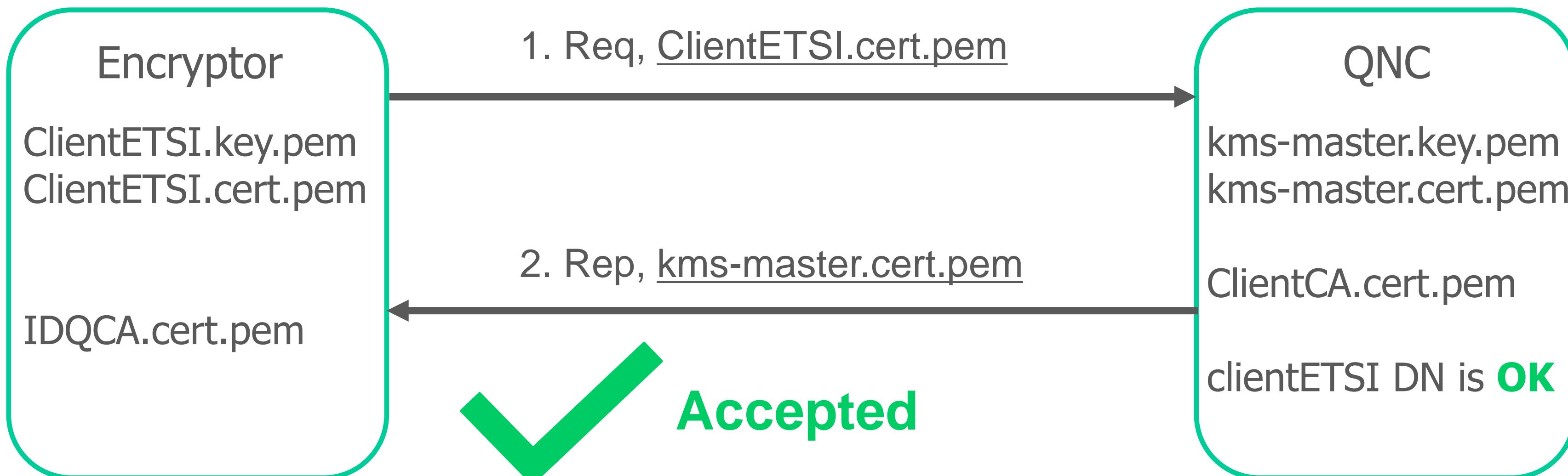
own:  
 ClientCA.cert.pem  
 ClientETSI.key.pem  
 ClientETSI.cert.pem



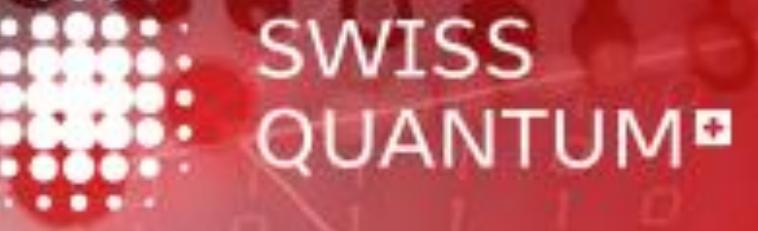
**IDQ**

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

Encryptor  
already  
store the  
root CA of  
IDQ



# A simple shell script to verify ETSI connectivity



```
#!/bin/sh

CURRENTDIR=$(dirname $0)

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

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

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

A diagram consisting of two green arrows. One arrow points from the text "Modify Key Interface Address, SAE Address, Cert and Key Locations" to the line "KMSM\_IP='192.168.10.101:443'". The other arrow points from the same text box to the line "Rep=\$(curl ...)".

# Output from shell script

Tue 09 Feb 2021 01:32:40 PM EST

Verified/Tested using Ubuntu20

Get key round

Get New Key from master

Found ID: ebf96e46-39b6-43c9-af1b-060d7a72bf2c

Key ID

Get Key with ID from slave

Found Key: mCJjc0Dask1MyxXMdf+UGDEQZxG7i6oMtD6g0JS01nA=

Quantum Key

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=

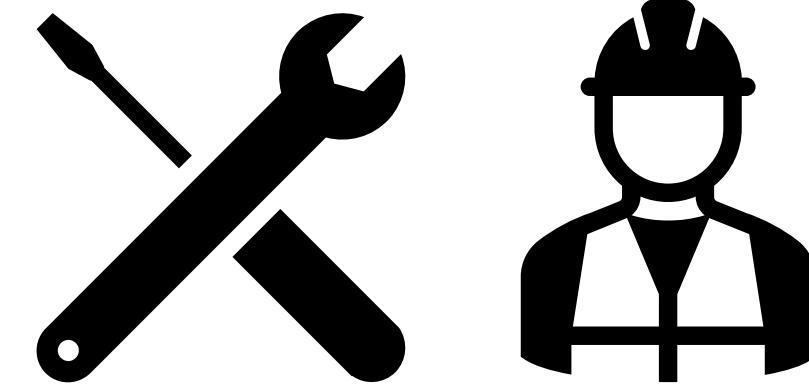
Found ID: 7e3ef78b-c14d-4bf7-a71a-8a2bbe36a305

Waiting 5 seconds for new request

# Troubleshooting

- The Cerberis XG service channel is not synchronizing (Service LED blinking):

1. De-connect the service channel fiber (with fixed attenuator if installed)
2. De-connect SFP module
3. Check the SFP insertion (in both terminals)
4. Clean the fixed attenuator connectors
5. Connect the fixed attenuator on the SFP, after the LC/UPC fiber is inserted a clear “click” must be heard.
6. Clean the service channel fiber connectors (in both terminals)
7. Connect the service channel fiber to the fix attenuator, after the LC/UPC fiber is inserted a clear “click” must be heard.



Note: If the problem persists check the attenuation on the service channel fiber (SFP are designed to work for 100km fiber distance, add fix attenuator to have between 12 to 20dB attenuation on the SC)

If during the key exchange a service channel ERROR will be shown in the logs check all the points above.

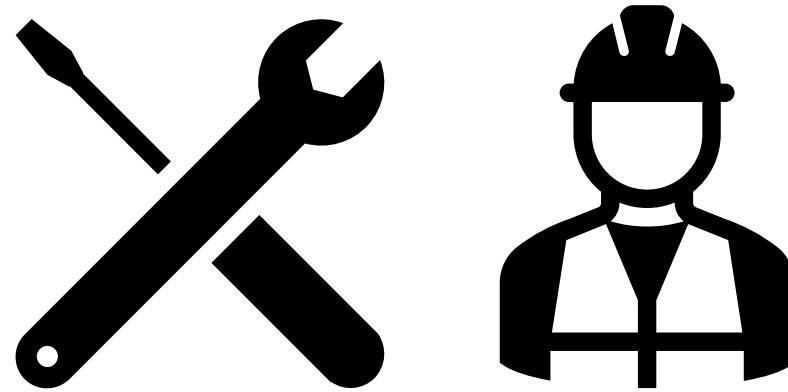
- Poor correlation during the “time bin alignment”

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

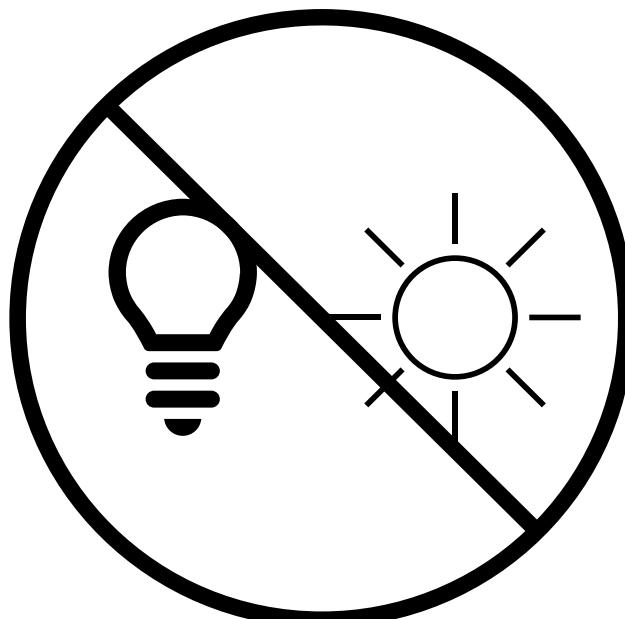
- System can arrive to “key exchanging” state, but QBER and Visibility are poor (QBER ~5%, Visibility ~90%)

1. Check the same points as above.

# Troubleshooting Cont'd



- Detection count is high during alignment or system unable to align quantum channel
- 1. Verify that excess light is not entering the fiber or fiber connector. Ambient light (especially sunlight) can penetrate most fiber patch cord shielding. Try turning off the lights, covering the fibers and connectors, or use armored fibers.



# Power Off



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

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

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

On Cerberis3: Log into the Shelf Manager. Enter the command: “chassis\_control -d” to shut down the chassis. The Blue LEDs will start to blink. Once the LEDs are solid blue it is safe to remove power.

# Factory Default

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

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

The following tasks are executed:

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

# Additional Commands



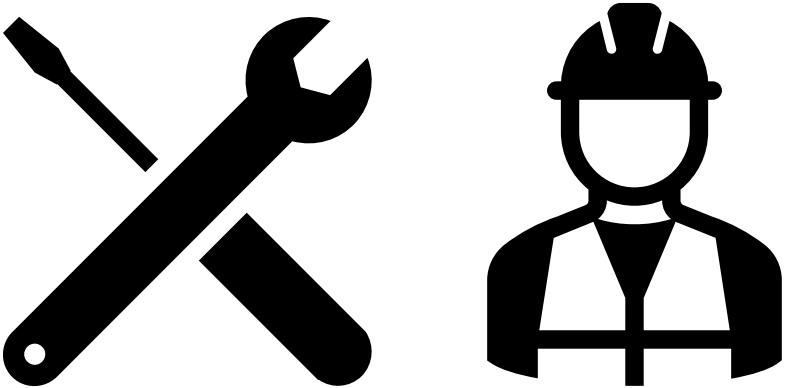
This training guide does not represent all the configuration/menu options on the CerberisXG. Please see the help menu on the QKD system for additional list of commands:

From the CLI, type the command “help”.

# Maintenance

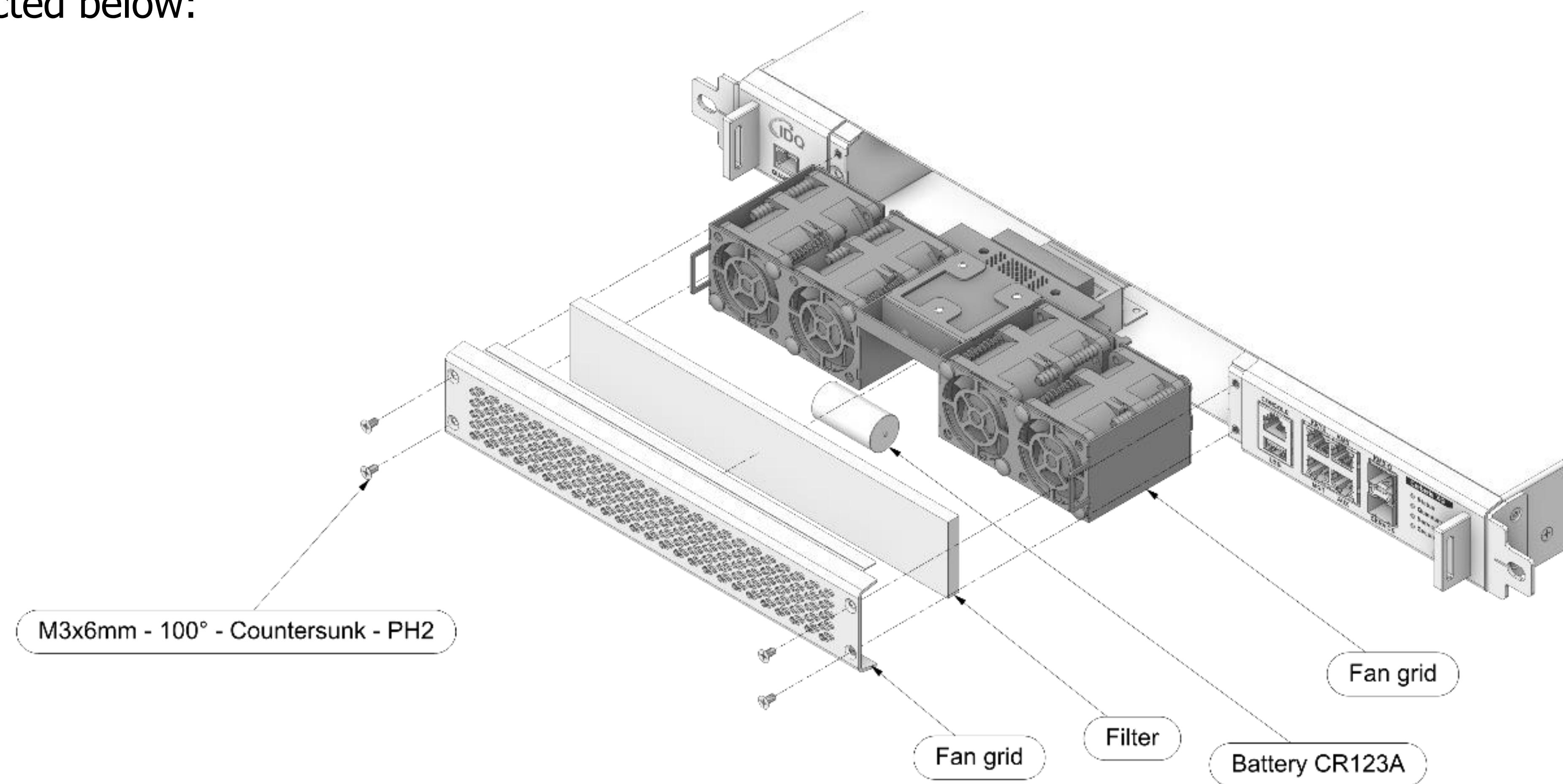
On your Cerberis XG, the following parts can be serviced in case of defect or replacement need:

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

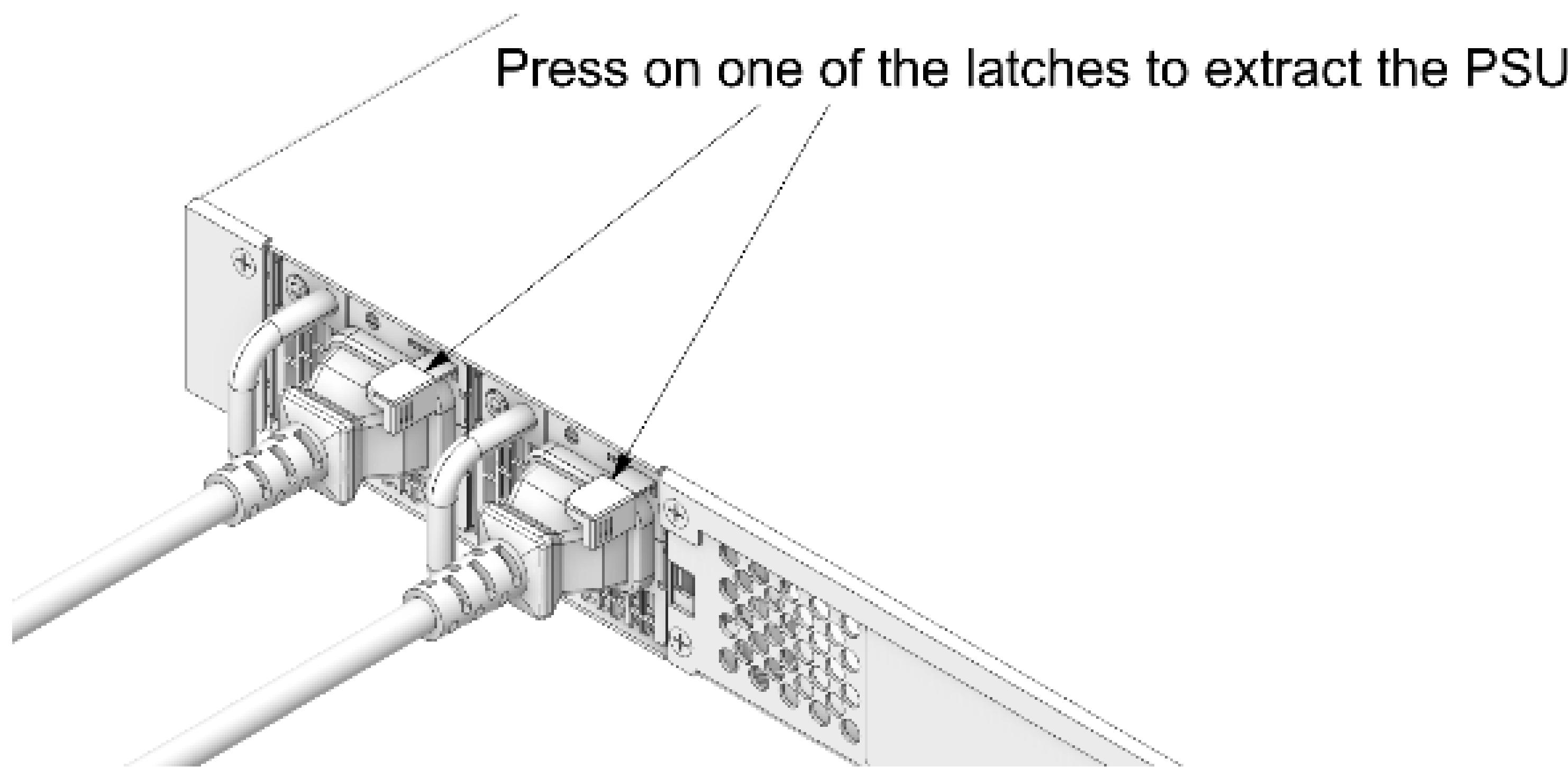


# Maintenance

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



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



# Upgrade Migration



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

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

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

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

# Research-Specific Functions

# Research-Specific Functions – Overview of steps



## Cerberis XGR/Clavis3

- Get/Set QKD Parameters

1. On QNET activate Research User Group :

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

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

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

1. On QNET Deploy Node Config :

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

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

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



# Authenticate as SuperAdmin First!



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

Enable the Research functionality:

qnet update role Research --enabled

Create a new Research user:

qnet create user testuser -r Research

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



## Re-Authenticate as the Research User



---

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

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

# Update Parameters



## Cerberis XGR

- Get QKD Parameters
  1. On QNET use **QKD show QKD** to show the QKD parameters:
    - qnet show qkd <QKD\_NAME> -url <QKD\_IP:8443>  
example :qnet show qkd QKDA –url <https://192.168.10.102:8443>  
or  
qnet show qkd QKDA
- Set QKD Parameters
  1. On QNET use **QKD update QKD** to edit the QKD parameters:
    - qnet update qkd <QKD\_NAME> <PARAM\_NAME> <VALUE>  
example : qnet update qkd NOMQKD /CommunicatorBob/FPGA/Model@Filter false

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

# Cerberis XGR/Clavis3 – Customizable parameters



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

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

**IDQ4P API – Raw Key Streaming**  
**See Supplemental Documentation**

# Thank You

