Digital Europe Programme Project **QCI-CAT**
QCI: Proof of Concept – Secure Connectivity Austria
Digital Europe Work Programme 2021-2022
EU Secure Quantum Communication Infrastructure (DIGITAL-2021-QCI-01)
Project number: 101091642

Project starting date: fixed date: 1 January 2023
Project end date: 30 June 2025
Project duration: 30 months

Document:	**Deliverable**
Type:	Report
Dissemination Level:	Public
Title:	**Report on HSM key-wrapping via QKD based VPN**
Work-Package	WP6
Document number:	D6.1
Document Owner:	CANCOM / Andreas Neuhold
Contributors:	AIT
Abstract:	This document outlines a use case designed to secure key material during transfer between two HSMs using a QKD and PQC based VPN.
Key words:	QKD, PQC, HSM, Quantum secure VPN
Pages	42
Delivery Date Planned	2025-02-28 (M3-30)

Revision History

Version	Revision Points	Author(s) & Organization	Date
0.1	Draft	A. Neuhold (CANCOM) G. Swoboda (CANCOM)	2024-10-17
0.2	Review Version	A. Neuhold (CANCOM)	2025-02-04
1.0	Final Version	A. Neuhold (CANCOM) G. Swoboda (CANCOM)	2025-02-17

Author List

Organization	Name	E-Mail address
CANCOM	A. Neuhold	andreas.neuhold@cancom.com
CANCOM	G. Swoboda	georg.swoboda@cancom.com

Reviewer List

Organization	Name	E-Mail address
TUG	Lena Heimberger	lena.heimberger@tugraz.at

Copyright Statement

Funding Acknowledgement:

This project has received funding from the European Union's Digital Europe Work Programme 2021-2022 under Project number 101091642 and the National Foundation for Research, Technology and Development.

Table of content

List of Figures

List of Tables

Executive Summary

This document describes a project aimed at securing Hardware Security Module (HSM) key material during transfers between two HSMs for backup or redundancy purposes. The project employs a Quantum Key Distribution (QKD)-based Virtual Private Network (VPN) to protect the data exchanged between HSMs. The infrastructure utilized includes secure data centers in Vienna, advanced network equipment, and state-of-the-art QKD devices. By integrating post-quantum cryptography (PQC) with QKD, the project ensures a high level of security, providing a scalable and efficient solution for cryptographic key management. The implementation of the developed key control software Arnika for WireGuard VPN ensures robust protection against potential threats, demonstrating the effectiveness of the proposed solution.

1. Introduction

In today's digital landscape, the security of cryptographic key material is paramount for organizations, governments, and military operations. This use case focuses on providing an additional layer of security to protect Hardware Security Module (HSM) key material during transfers between two HSMs for site-redundant backup or synchronization tasks, ensuring robust protection of sensitive cryptographic material. The key material is essential for tasks such as certificate generation, verification, and signing operations, which serve as the root of trust for critical IT infrastructure.

To address the challenges posed by vendor proprietary solutions and enhance security, the use case proposes utilizing a Quantum Key Distribution (QKD)-based Open Source Virtual Private Network (VPN). This approach aims to make it impossible for attackers to access the data exchanged between HSMs. Additionally, a Demo App has been developed to emulate typical IT cryptographic operations, such as encrypting and decrypting messages using the key material stored in the HSMs and verifying exchanged messages, further demonstrating a real world application of the proposed solution.

The use case was successfully set up and ran between 02.02.2024 and 27.06.2024

1.1. Purpose and scope of the document

The purpose of this document is to provide a comprehensive overview of the use case aimed at securing HSM key material during transfers between two HSMs. It details the objectives, infrastructure, network design, and the development and implementation of a quantum secure VPN. The document also covers a Demo App to demonstrate the secure transfer and verification of key material between HSMs. The scope includes the technical specifications, methodologies, and security measures employed to achieve the use case’s goals.

1.2. Target Audience

This document targets network engineers and the general public interested in deploying quantum technologies in their networks.

1.3. Relation to other project work

Immediate relation to the WP6.

1.4. Structure of the document

- **Executive Summary**: A brief overview of the use case objectives, key components, and outcomes.
- **Introduction**: An introduction to the importance of securing cryptographic key material and the use case approach to addressing this challenge.
- **Purpose and Scope of the Document**: Explanation of the document's purpose and the scope of the use case.
- **Objectives**: Description of the use case's goals and the importance of securing HSM key material.
- **Requirements**: Identification of the infrastructure and network requirements for implementing the use-case.
- **Quantum Secure VPN**: Discussion of the choice of WireGuard over IPSEC, the development of Arnika, and the integration of PQC with QKD.
- **Demo Application**: Overview of the Demo App developed to demonstrate the secure transfer and verification of key material between HSMs.
- **Infrastructure**: Description of the data centers, network equipment, and additional hardware used in the use case.
- **Network Design**: Explanation of the network development process, including network security zones, VLAN (Virtual Local Area Network) and IPv4 subnet demarcation, and the network connectivity matrix.
- **Appendices**: Additional information, including a list of acronyms and a bibliography.

2. Objectives

The objective is to secure HSM (Hardware Security Module) key material during transfers between two HSMs for backup or redundancy purposes. The key material is crucial for sensitive cryptographic operations, including certificate generation, verification, and signing operations for firmware and other critical IT infrastructure tasks. Such cryptographic material serves as the root of trust for organizations, governments, and military operations.

The solution aims to utilize key-wrapping techniques to transfer sensitive key material between HSMs, while addressing the challenges posed by vendor proprietary solutions that are difficult to audit and verify. To enhance security, the proposal suggests protecting the link between two HSMs with a QKD-based VPN, making it impossible for attackers to access the data exchanged between the HSMs.

A Demo App should emulate typical IT cryptographic operations, such as encrypting and decrypting messages using the key material stored in the HSM.

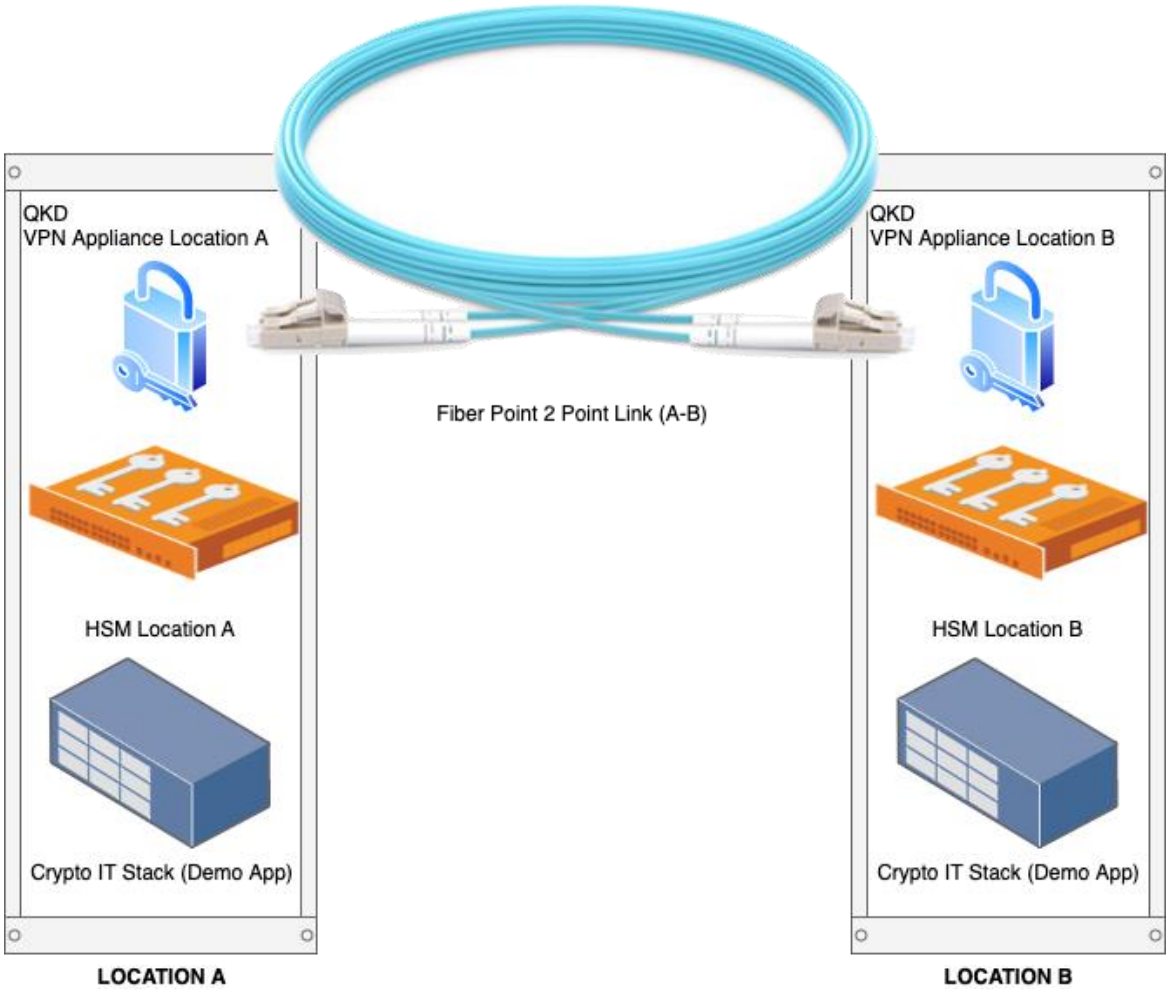


Figure 1- High-level overview of the proposed HSM transfer concept.

3. Requirements

The following requirements have been identified for the implementation of the use-case:

Infrastructure, which should meet all the criteria for a QKD setup, including a secure data center, dark fibers for quantum channel and proprietary protocols, as well as conventional network infrastructure. The data centers we have chosen for the use-case were one of our company own data centers located at Euro Plaza in Vienna and one of our partner's NTT data center called "Vienna 1". Refer to the NTT website for more information about their data center services https://services.global.ntt/en-us/services-and-products/global-data-centers/global-locations/emea/vienna-1-data-center. We were provided with an upstream internet connection, including a public IP subnet that spanned across the two data centers. This setup allowed us to build the environment for our use case.

Network, starting with network design that is based on a given functional architecture and realized with proper network equipment such as switches, routers and firewalls. The MikroTik CCR2116-12G-4S+, featuring a powerful 16-core ARM CPU, was used as the core network component in each data center. This router offers the most features and performance for the money, including four 10G SFP+ ports and twelve Gigabit Ethernet ports. This specific model from MikroTik not only offers switching but also routing, firewalling and VPN capabilities which makes it possible to create a cost-effective full network stack just with a single network device. For full specifications, please refer to the vendor's product homepage: https://mikrotik.com/product/ccr2116_12g_4splus. It is worth noting that MikroTik is headquartered in Riga, Latvia. This makes them a vendor within the European Union, which is an important consideration for compliance and regulatory purposes.

Figure 2- The MikroTik CCR2116-12G-4S+ router used as the core network component in each data center.

As **Hardware Security Modules (HSM)**, two entry-level models of the Thales network HSM series, the Luna A700, have been chosen. This is a model of the Luna network HSM series designed for high performance cryptographic processing and key management. The Luna A700 HSM offers standard performance with 2 MB of memory and can be divided into up to 5 partitions, each with independent data, access controls, and administrative policies. This model is validated to FIPS 140-2 Level 3 standards, providing a high level of security for sensitive cryptographic data transactions. It is also designed to resist tampering, ensuring the integrity and security of cryptographic keys. For more detailed specifications please refer to the Thales product page https://cpl.thalesgroup.com/encryption/hardware-security-modules/network-hsms. The two HSM are configured for site redundancy and high availability purpose to be used on different locations and to be synced/backup over quantum secure VPN tunnel.

The **QKD** devices for this use-case has been provided by the AIT. It turned out to be the Cerberis XG QKD system from the vendor IDQ - ID Quantique. For more detailed information please refer to the Vendors product page https://www.idquantique.com/quantum-safe-security/products/cerberis-xg-qkd-system/

In terms of additional **hardware**, beside the QKD and HSM devices, two of conventional 19-inch server hardware has been used to host all the necessary software. This included the DEMO application, the

VPN which functioned as a software encryptor, the HSM client, and other necessary software. The combination of several functions installed in virtual machines on a single hardware per data center ensures cost efficiency. The specific hardware model of the server that has been used for the use-case is **HPE ProLiant DL160 Gen10 server**. Please refer to the vendor product page for detailed specifications https://www.hpe.com/psnow/doc/a00021860enw.

The **software stack** comprises conventional open-source operating systems (Ubuntu 22.04.3 LTS) and open-source frameworks (OpenSSL3.02, php8.1, nginx1.18) and toolsets (tmux, WireGuard, Rosenpass). In addition, proprietary software and drivers are included, bundled and shipped with the HSMs and QKD devices, which are required for their operation.

4. QKD

The QKD hardware has been provided, installed and configured by the experts from AIT.

The Cerberis XG is IDQ's fourth generation of QKD systems, designed for easy installation and high availability in data centers and telco infrastructures. It features a compact 19'' rackmount 1U size, hot-swappable components, and minimal onsite support requirements.

Key capabilities include:

- Medium-range interconnection (up to 90 km)
- Standard key rate of 14,000 AES-256 keys per hour at 18 dB
- Support for complex network topologies
- Centralized control and monitoring
- Interoperability with major Ethernet and OTN encryptors

4.1. QKD metrics

QKD systems like the Cerberis XG from ID Quantique provide a comprehensive monitoring and visualization platform. The platform enables to track critical metrics, central configuration, and integration with network management protocols.

During the use-case the most essential metrics has been collected and visualized to ensure the security and performance of the quantum communication channel to quickly identify and address potential issues, ensuring the reliable operation of the QKD infrastructure.

4.1.1. QKD key rate

The **key rate** is a fundamental parameter in QKD, determining the feasibility and security of the quantum communication system. It refers to the rate at which secure cryptographic keys are generated and distributed between two parties. This rate is crucial for the efficiency and practicality of QKD systems. The key rate can be influenced by various factors, including the quality of the quantum channel, the presence of noise, and the specific QKD protocol used.

Figure 3 – Graph of the Keyrate

4.1.2. QKD Quantum Bit Error Rate

The **Quantum Bit Error Rate (QBER)** is a crucial parameter in QKD systems, ensuring the security of the quantum channel and identifying potential impairments or eavesdropping attempts. It represents the probability of a mismatch between the signals sent by the sender (Alice) and those received by the receiver (Bob).

QBER for good operation should be below 5%. A QBER above this threshold can still allow for secure key generation but may require more error-correction. If it exceeds a theoretical limit (approximate around 11%), the channel is considered compromised by eavesdropping, making secure key generation impossible. QBER can be influenced by factors such as environmental noise, device imperfections, and transmission losses. In satellite-based QKD, QBER is affected by atmospheric losses, stray light, polarization errors, and detector noise.

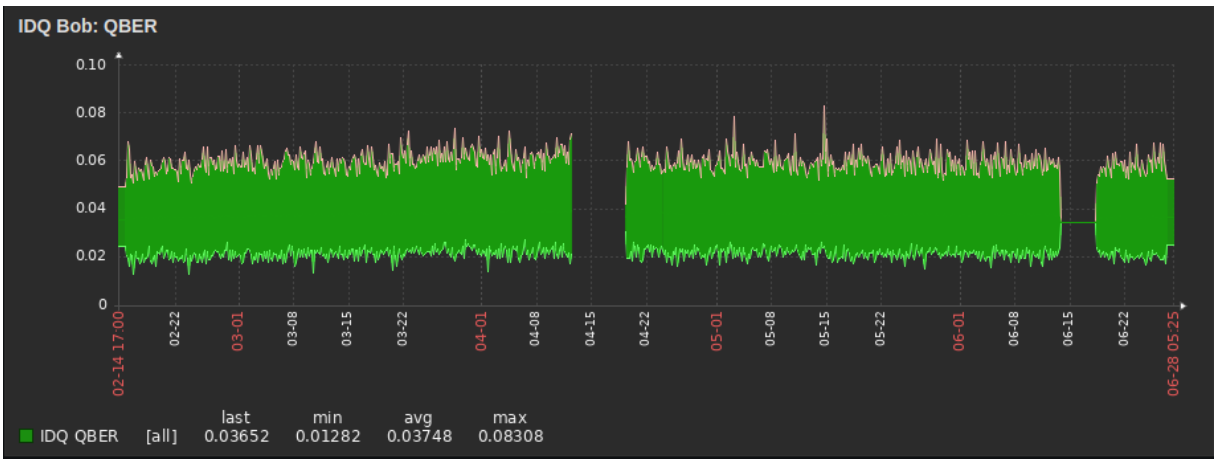


Figure 4 – Graph of QBER (Quantum bit error rate)

4.1.3. QKD visibility

QKD **visibility** is a critical parameter that measures the interference between pulses in different time bins. High visibility, typically above 97%, indicates minimal disturbance or interference ensuring the secure transmission of quantum signals. QKD protocols rely on the interference of photons. Visibility refers to how clearly these photons interfere. High visibility means the photons interfere strongly, producing a clear pattern of constructive and destructive interference and it indicates that the quantum states of the photons are well-preserved during transmission and not significantly disturbed.

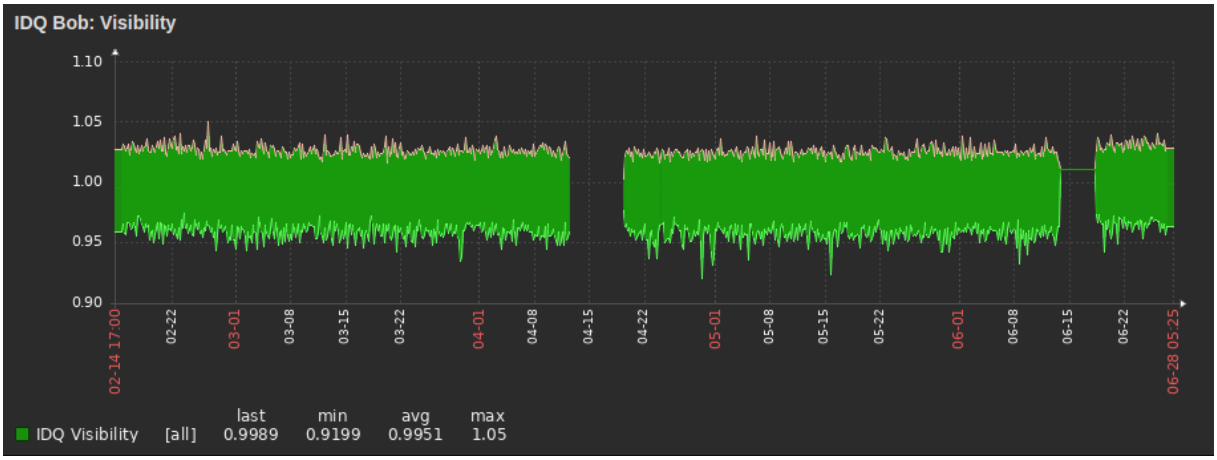


Figure 5 – Graph of QBER (Quantum bit error rate)

5. HSM – Hardware Security Modules

Hardware Security Modules (HSM) are physical devices that provide secure generation, storage, and management of cryptographic material.

The HSMs used in this use case are Thales Luna Network HSM model Luna A700 version LunaSA 7.8.4.

This section describes the steps to set up two HSMs (qcicat-o93-hsm01 and qcicat-o73-hsm02) so that the two application servers (qcicat-o93-app01 and qcicat-o73-app02) can communicate with the HSMs for cryptographic operations. The setup of HSMs involves several steps, including initial configuration, partition creation, client assignment, and network setup.

The solution must be set up so that all communications between a client and HSM, and the communications (sync/backup) between two site-redundant HSMs, are secured by the Quantum secure VPN tunnel.

Prerequisites

- Thales Luna Network HSM appliances (Luna A700 with firmware version LunaSA 7.8.4)
- Client machines running Ubuntu Linux 22.04.3 LTS
- Required software for HSM: Luna Client, **lunacm**
- Local and SSH access to HSMs and client machines

Prepare HSMs

Recover the Thales NET HSMs from Secure Transport Mode (STM) to verify integrity before initialization.

Initialization of the HSM

Connect via console cable, start the initialization process, set Security Officer (SO) and Domain Passwords (Cloning Domain), and confirm initialization status.

Network Configuration

Configure network settings (IP, netmask, gateway, DNS) for HSMs and verify connectivity.

Initial Configuration of HSMs

Connect to HSMs via SSH, log in as admin, and initialize with specific labels (e.g. **hsm01** and **hsm02**) and passwords.

Partition Creation

Create two partitions **qci_cloning** and **qci_export** on the HSMs to logically separate cryptographic operations and key storage.

Partition Initialization

Set Partition owner's password and Domain name and configure partition policies.

Client Assignment

Assign the created partitions to client machines (**app01** and **app02**) and deploy client configuration.

With these steps the setup of Luna Network HSMs and application servers was completed. The steps outlined in this report can serve as a guide for future HSM setups. For detailed steps refer to the Thales Luna documentation which is online available on The Thales Support Documentation website:

https://thalesdocs.com/gphsm/luna/7/docs/network/Content/Configure.htm
https://thalesdocs.com/gphsm/luna/7/docs/network/Content/admin_appliance/network_config/appliance_config.htm

6. Quantum secure VPN

One of the core key components of the use-case is the quantum secure VPN connection.

As a preliminary measure, an investigation was conducted into the available VPN software. The investigation focused on those distributed as open-source software that offered already post-quantum secure encryption or the possibility to add such a feature.

An interesting finding when investigating in IPSEC solutions shows that a number of methodologies for the implementation of post-quantum encryption for IPSEC have been patented.

- US: https://patents.google.com/patent/US7602919B2/en
- CN: https://patents.google.com/patent/CN101142779A/en
- EU withdrawn: https://patents.google.com/patent/EP1864417A4/en

It seems that major corporations are engaged to secure patents in the adoption of IPSEC with post-quantum secure measures and establishing competing standards (e.g. RFC 8784 , RFC 9370)

However, in the use-case we favored WireGuard over IPSEC due to its simplicity, efficiency, and modern cryptographic design.

WireGuard is designed to be lightweight and user-friendly, with a codebase significantly smaller than IPSEC, which reduces the risk of vulnerabilities and simplifies audits. This streamlined design makes it more adaptable to custom use-cases, such as integrating QKD or PQC-generated keys, without the need to deal with complex configurations or legacy support issues often associated with IPSEC.

Considering the flexibility characteristics of WireGuard and in accordance with the contributions of the **Rosenpass** team who have added a PQC functionality to WireGuard by injecting symmetrical keys as pre-shared keys into WireGuard it was obvious to choose WireGuard as VPN function for the use-case.

Rosenpass is free and open-source software based on the latest research in the field of cryptography. It is intended to be used with WireGuard VPN but can work with all software that uses pre-shared keys. It uses two cryptographic methods Classic McEliece and Kyber to secure systems against attacks with quantum computers. Please refer to the Rosenpass website https://rosenpass.eu/ for more detailed information.

It is worth mentioning, that the initial idea was to use **Rosenpass** in the first place. A feature request has been submitted to the **Rosenpass** GitHub project, requesting the incorporation of an ETSI014 key handling feature into Rosenpass. After careful consideration, it was determined that implementing such a change in **Rosenpass** was not feasible, particularly within the available timeframe.

Therefore, we developed the missing **ETSI014** feature as a standalone key control function on our own. With support and inspiration from the **Rosenpass** team we have followed up and pursued a similar approach. This standalone key control function is capable of injecting a symmetrical key into WireGuard similar to **Rosenpass**, with the difference that the key is provided by a QKD via the **ETSI014** interface. In addition, we implemented also an option to use a PQC key provided by **Rosenpass** as well as an option to use a HKDF key derivation of both QKD and PQC key**.** This "key management function" has given the name **Arnika** and has been published on Github https://github.com/arnika-project/arnika under open-source license Apache 2.0.

6.1. Concept

The high-level concept signifies a layered approach to secure data transmission, integrating post-quantum cryptography and quantum key distribution technologies.

This approach is illustrated in Figure 6, which shows a pipe representation of layered security measures for protecting data transfer using WireGuard VPN with PQC and QKD.

Our objective was to ensure that the solution remains quantum secure even if one of the employed technologies QKD or PQC is compromised.

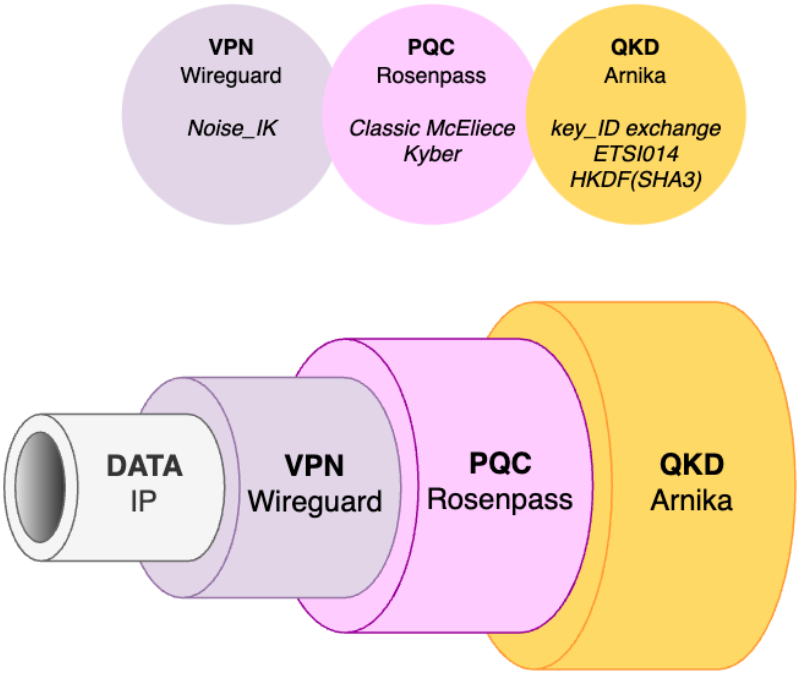


Figure 6 – High level representation of mixed QKD and PQC encryption concept in pipe view

6.2. WireGuard

WireGuard uses state-of-the-art cryptography and secure trusted constructions. It makes conservative and reasonable choices and has been reviewed by cryptographers.

It provides faster connection speeds and lower latency due to its lightweight architecture and efficient use of cryptographic methods. As an open-source project, it allows users to inspect, modify, and improve its codebase, enhancing transparency and trust. Its minimalistic design ensures low resource consumption while maintaining high performance, making it suitable for both personal and enterprise use.

WireGuard's modern cryptographic primitives have been designed to be secure against contemporary threats and adaptable to post-quantum cryptographic upgrades. The additional layer of symmetric-key crypto for post-quantum resistance is available in the form of an optional pre-shared key that is mixed into the public key cryptography.

As per WireGuard protocol documentation (https://www.wireguard.com/protocol/#primitives) the following protocols and primitives are used:

- ChaCha20 for symmetric encryption, authenticated with Poly1305, using RFC7539's AEAD construction
- Curve25519 for ECDH key exchange
- BLAKE2s for hashing and keyed hashing, described in RFC7693
- SipHash24 for hash table keys
- HKDF for key derivation, as described in RFC5869

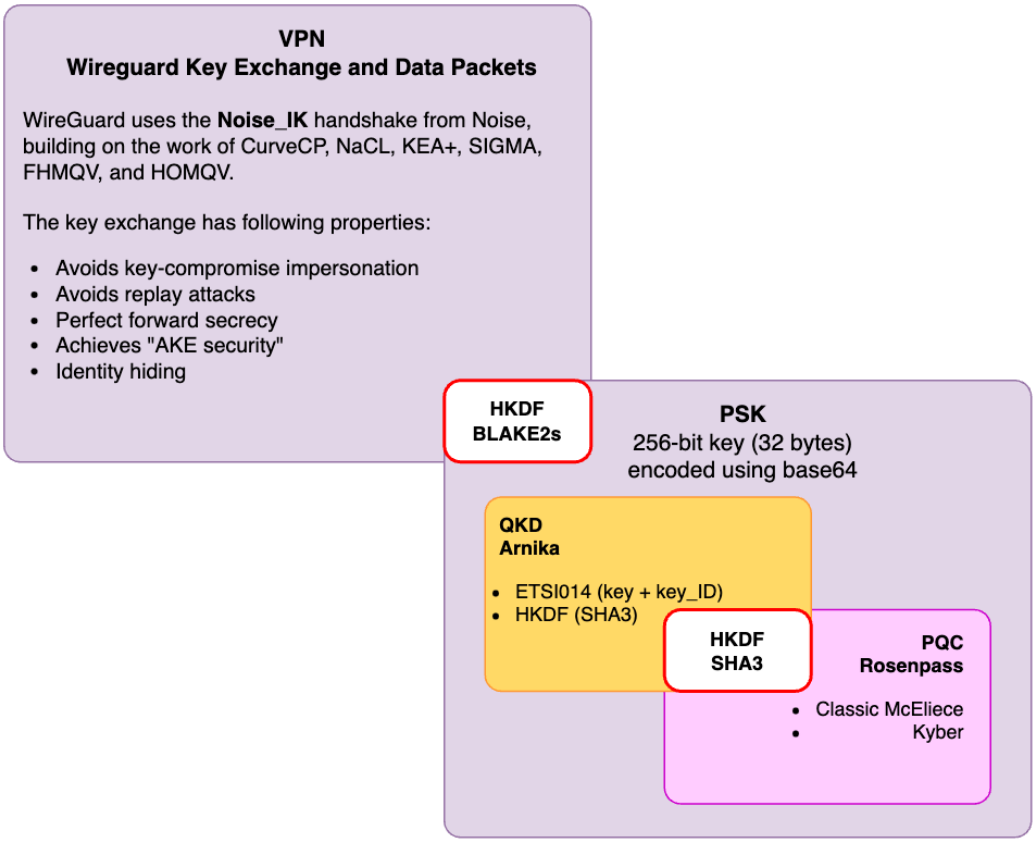


Figure 7 – WireGuard cryptographic properties and primitives, including QKD and PQC key material as pre-shared key.

The Figure 7 is visual representation of WireGuard's cryptographic primitives, illustrating its key exchange mechanisms and integration with pre-shared key for symmetric key cryptography. WireGuard employs the Noise_IK handshake protocol, leveraging various cryptographic frameworks to deliver essential security features such as resistance to key-compromise impersonation, replay attack prevention, perfect forward secrecy, authenticated key exchange (AKE), and identity hiding. Additionally, it depicts a 256-bit pre-shared key encoded in base64, derived from PQC and QKD system using HKDF. The PQC key is provided by **Rosenpass**, utilizing schemes like Classic McEliece and Kyber, while the QKD key is provided by **Arnika** via gathering key material via ETSI014 from the QKD KMS (Key Management System).

For more details information and analysis please refer to following references:

Jason A. Donenfeld: **WireGuard: Next Generation Kernel Network Tunnel**
https://www.wireguard.com/papers/wireguard.pdf

Dowling/Paterson: **A Cryptographic Analysis of the WireGuard Protocol:**
https://www.wireguard.com/papers/dowling-paterson-computational-2018.pdf

Kudelski Security: **Adding quantum resistance to WireGuard**:
https://research.kudelskisecurity.com/2021/07/08/adding-quantum-resistance-to-wireguard/

6.2.1. Pre-shared key injection into WireGuard

As already mentioned, and documented on the online WireGuard Protocol documentation under **Key Exchange and Data Packets** in the section **Key Exchange and Data Packets** https://www.wireguard.com/protocol/#key-exchange-and-data-packets , there is a built-in pre-shared key feature. If it is not used, the 256-bit pre-shared key defaults to all zeros.

If an additional layer of symmetric-key cryptography is required (for, say, post-quantum resistance), WireGuard also supports an optional pre-shared key that is mixed into the public key cryptography. When pre-shared key mode is not in use, the pre-shared key value used below is assumed to be an all-zero string of 32 bytes.

A pre-shared key can be provided in the configuration file, which is rather static and valid for the entire lifetime of the VPN tunnel. However, there are ways to inject a pre-shared key on runtime via WireGuard tools or netlink interface, also there are functions available for various programing languages.

The WireGuard **wg** command line option for injecting a pre-shared key during runtime:

```
wg set <interface-name> peer <peer-public-key> preshared-key <base64-encoded-key>
```

Example of a command line options for a shell script to inject a base64 encoded key pre-shared key located in a file **<file-with-base64-encoded-key>** into a WireGuard device **<interface-name>** for the peer **<peer-public-key>**

```
wg set <interface-name> peer $(cat <file-with-base64-encoded-key>) preshared-key
<(echo <peer-public-key>
```

For the full option set of the **wg** command line utility, please refer to the man page or the related WireGuard online documentation https://git.zx2c4.com/wireguard-tools/about/src/man/wg.8

The key injection in our software **Arnika** (which is written in Golang), is realized by using the related Golang functions (golang.zx2c4.com/wireguard/wgctrl) provided by WireGuard. The required functions are available in the official repository. Please find a list of repositories for various projects including tools and implementations available for various operating systems and programming languages, on the WireGuard website https://www.wireguard.com/repositories/

The injection of a new symmetric key as pre-shared key should be done periodically, preferably before each rekey event. The rekey process in WireGuard is designed to ensure the continuous security of the VPN connection by periodically refreshing the cryptographic keys. WireGuard initiates a new handshake to rotate the encryption keys based on specific conditions. Typically, WireGuard will rekey after 120 seconds or after 2^{60} messages, depending on which condition is met first. This dual approach

ensures that the keys are regularly updated, maintaining the security of the connection and providing perfect forward secrecy.

The re-keying process is seamless and does not disrupt ongoing communication, as WireGuard handles it efficiently in the background.

In **Arnika**, the interval for key gathering and injection into WireGuard is configurable, but it is recommended to set it to 120 seconds (default value) to align with the WireGuard rekey process.

NOTE: When no data passes the WireGuard interface, then there will be no key exchange triggered by WireGuard!

6.3. Arnika

Arnika is a compact, lightweight external standalone extension for WireGuard VPN, engineered to incorporate symmetric keys as pre-shared keys into WireGuard. By adding such an additional significant layer of security this integration ensures the establishment of a quantum-secure VPN (safeguarding against compromise of session keys).

Arnika gathers a 256-bit symmetric encryption key from a KMS within a QKD infrastructure, shares the associated key_ID with an Arnika peer, and then both Arnika instances symmetrically configure a pre-shared key for WireGuard using the obtained key material.

Furthermore, Arnika offers an additional security layer and integrate PQC by leveraging key material provided by a PQC system like **Rosenpass**. This PQC key is then mixed with the QKD key via HKDF to create an even stronger pre-shared key for WireGuard. This pre-shared key then benefits from both PQC and QKD, offering enhanced protection.

Arnika is available on Github https://github.com/arnika-project/arnika and licensed under open-source license Apache 2.0.

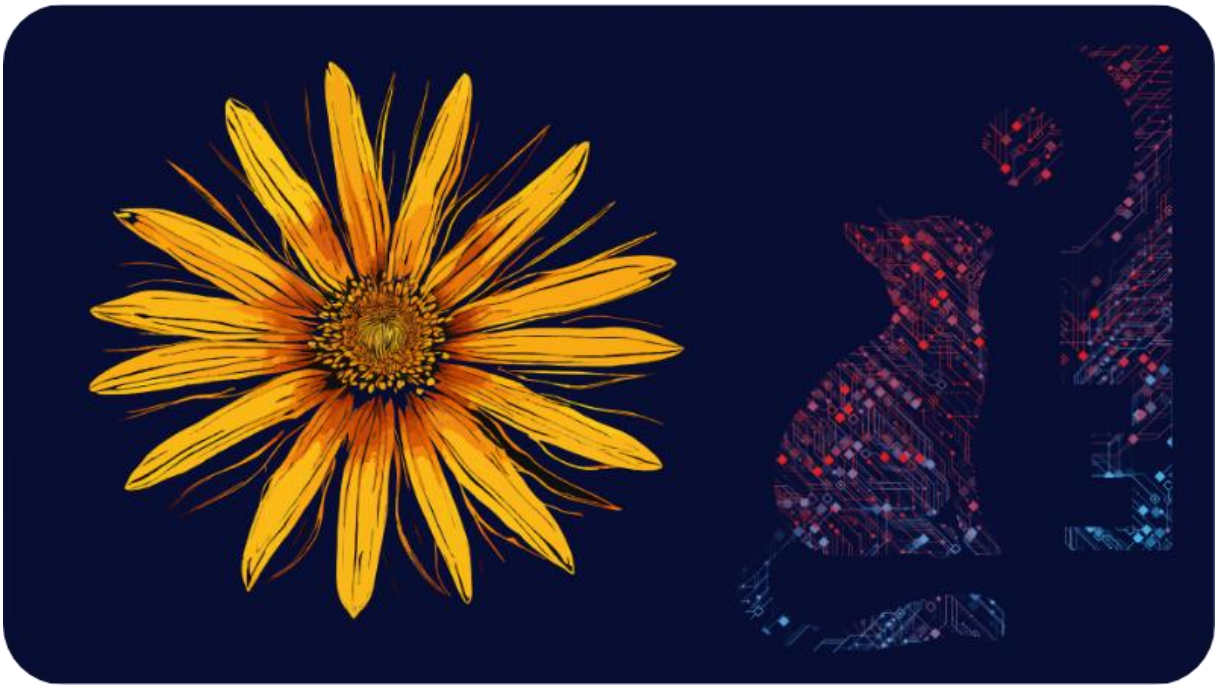


Figure 8 – Arnika Logo

6.3.1. Arnika functionality

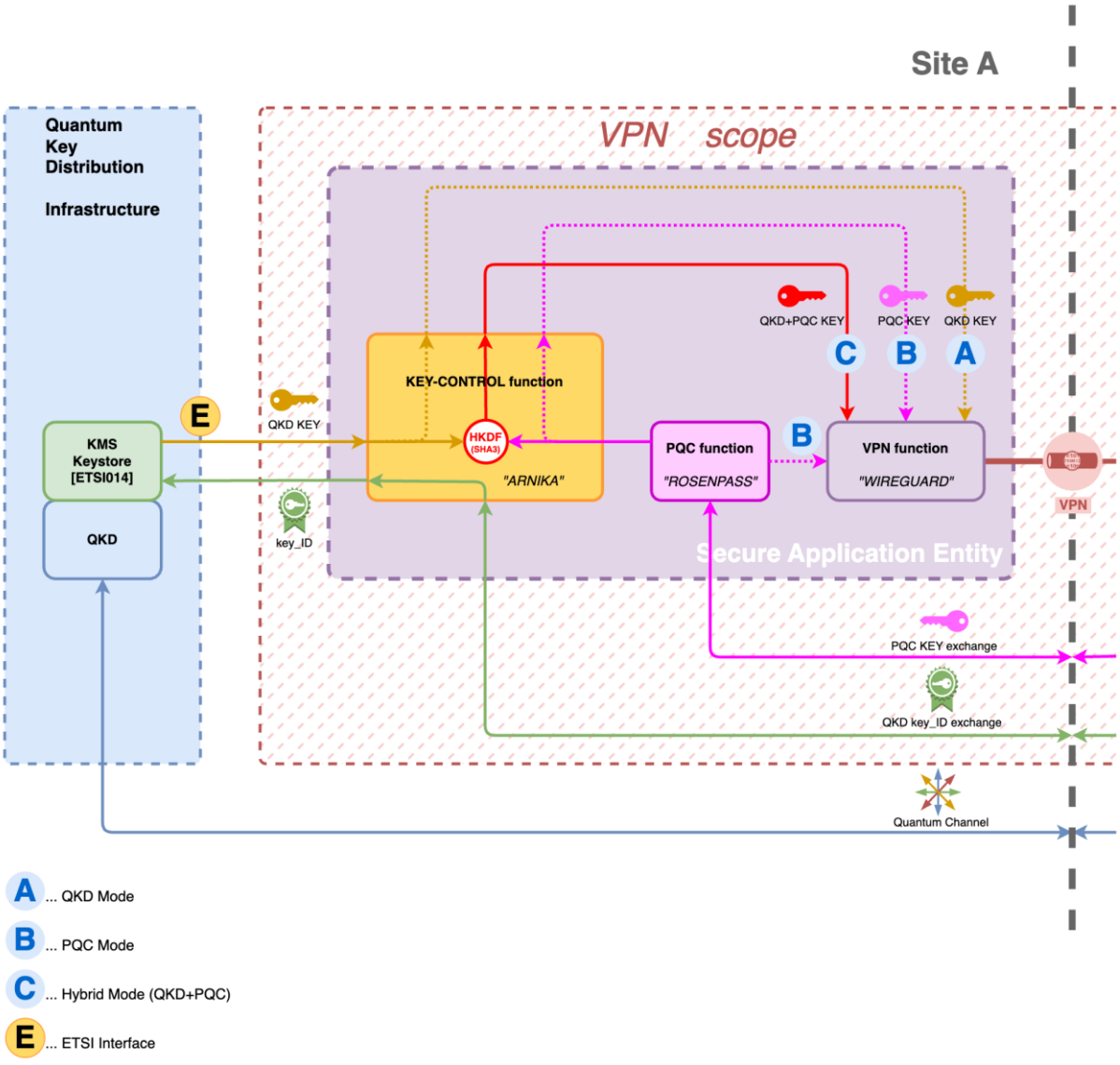


Figure 9 – VPN functional overview in flow representation

As illustrated in Figure 9, **Arnika** support 3 ways of controlling key material: QKD mode (A), PQC mode (B), and mixed mode QKD+PQC (C). The process for controlling the key material for scenario (C) (QKD+PQC) is outlined as follows:

The **KEY-CONTROL function (Arnika)** serves as a control entity, responsible for obtaining a **key** and transferring it to the **VPN function** (WireGuard). The QKD key is obtained via ETSI014 interface (E) from the QKDs embedded KMS and the PQC key is obtained via API or pointer/file descriptor from the **PQC function** (**Rosenpass).** This can be any alternative or already existing PQC function/implementation which is capable to provide PQC key materials in 256bit - base64 encoded key format. Subsequently, the **KEY-CONTROL function** derives a single key from the two input keys **QKD key** and **PQC key** by using a **HKDF** key derivation function with SHA3-256 hash function (QKD. The specific derivation function, whether **HKDF** or an alternative, is a topic open for discussion among cryptographic experts.

6.3.2. Key derivation

For the key derivation of the QKD and PQC key, **HKDF** was chosen due to its efficiency and security and its ability to generate strong, unique keys from potentially weak or variable input material. It ensures that even if a key is compromised, the derived keys remain secure.

HKDF is a key derivation function based on the HMAC message authentication code described in
- RFC 5869 (https://www.rfc-editor.org/rfc/rfc5869.txt) and
- NIST SP800-56Cr2 (https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Cr2.pdf)

WireGuard uses **HKDF** with the BLAKE2s hash function for its key derivation process. This combination is part of the Noise protocol framework, which WireGuard is built upon. **HKDF** is used to derive strong encryption keys from the initial key material generated during the key exchange phase.

Referring to the noise protocol documentation https://noiseprotocol.org/noise.html#pre-shared-symmetric-keys the **MixKey()** design uses **HKDF** because:

- HKDF is well-known and HKDF "chains" are used in similar ways in other protocols (e.g. Signal, IPSEC, TLS 1.3)
- HKDF has a published analysis (https://noiseprotocol.org/noise.html#ref-hkdfpaper)
- HKDF applies multiple layers of hashing between each **MixKey()** input. This **extra** hashing might mitigate the impact of hash function weakness.

Ultimately, both **Arnika** and WireGuard are using HKDF (despite difference hashing function) for mixing cryptographic key material. As a result, two **HKDF** are taking place sequentially with a combination of three keys generated by different cryptographical primitives, from conventional to PQC and QKD.

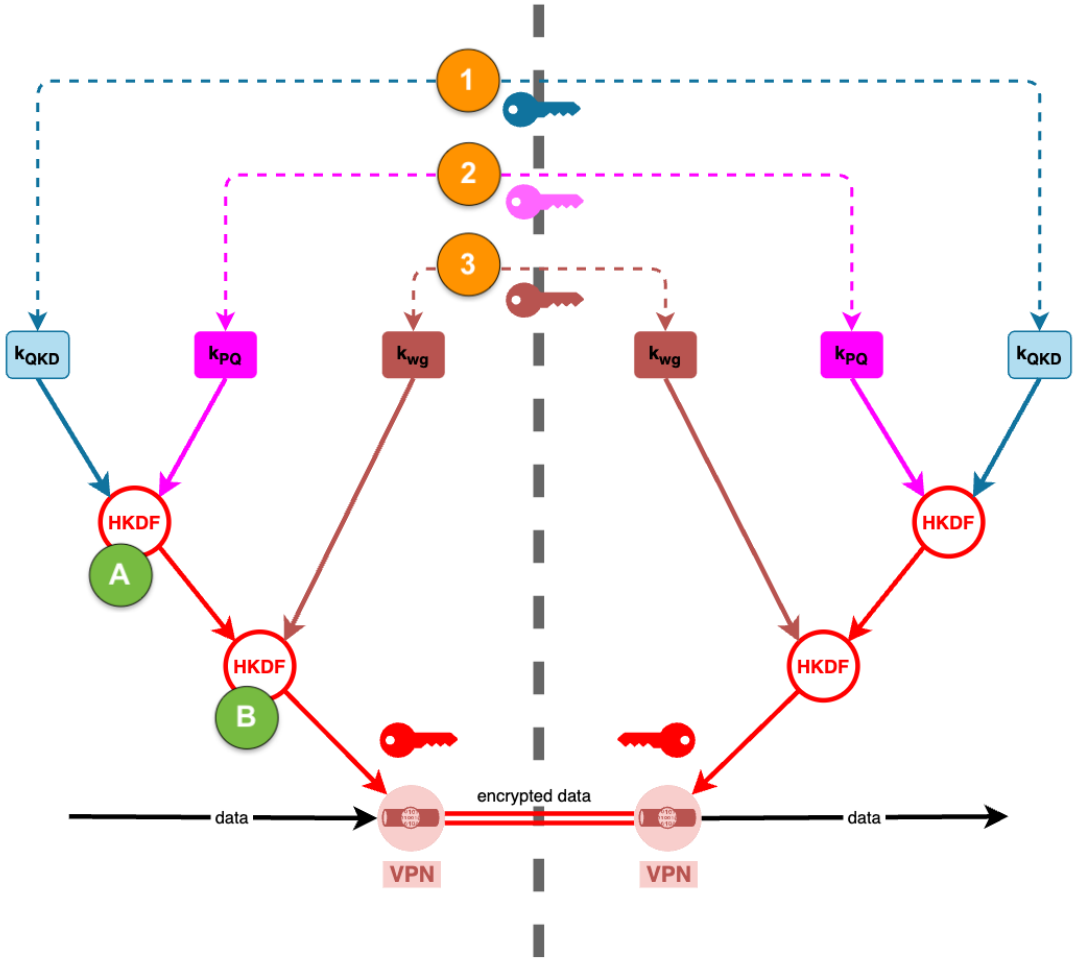


Figure 10 – Key derivation tree representation.

1. Key generated by a **QKD** system and provided via ETSI014 from a KMS
2. Key provided from a **PQC** system like **Rosenpass** which was generated via **PQC** key exchange
3. Key from conventional key exchange done by **WireGuard**

A. HKDF + SHA3 key derivation with QKD and PQC key

B. HKDF + BLAKE2s key derivation with the product of A. (QKC+PQC) and a conventional **WireGuard** key

6.3.3. Arnika internals and mode switching

As shown in Figure 11, Arnika works with asynchronous functions, that is why it is almost impossible to visualize the internal messages/processes in a traditional sequence diagram.

Keeping the peers synchronized is implemented in an unconventional way. It is based on some simple but effective rules, such as *whichever peer receives a key_id first becomes the passive backup peer for the configured interval*. It also results in a bare minimum of intercommunication as well as prevention of shared brain situation. Ultimately, any event that causes a rule to fail will automatically result in a different pre-shared key and thus a non-working WireGuard VPN connection.

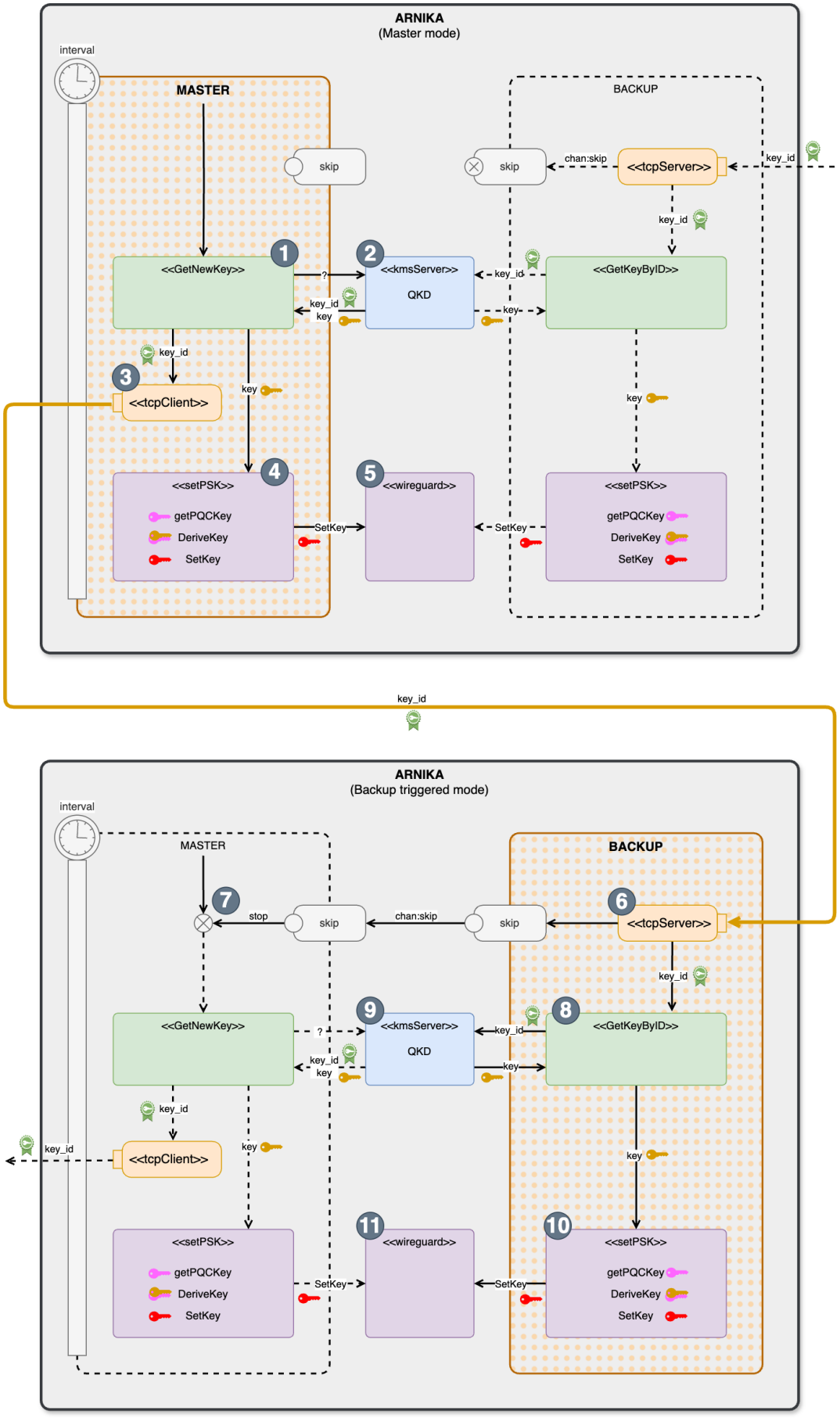


Figure 11 – Arnika internals – synchronization and mode switching

Arnika Master mode: The first action that is happening when starting Arnika is requesting a key from the KMS (1), (2). After Arnika receives a **key** and the related **key_id** from the KMS, Arnika is sending the **key_id** to its peer (3) and the **key** to the pre-shared key function (4) to inject the **key** (depending on the configuration QKD or QKD+PQC HKDF) into WireGuard (5)

Arnika Backup mode: The Arnika peer that receives a **key_id** (6) is becoming immediately a backup instance (7) for the next period of the configured interval. In Backup mode Arnika is using the received **key_id** to request the related key from the KMS (8)(9). The received **key** will be sent to the pre-shared key function (10) to inject the **key** (depending on the configuration QKD or QKD+PQC HKDF) into WireGuard (11).

6.3.4. Routing

As the application server hosts several functions like Arnika (QKD), Rosenpass (PQC), the lunacm (HSM client) and WireGuard VPN (as the encryptor), the routing is a bit special.

The HSMs needs to be configured in a way that the related application server that is hosting the WireGuard VPN acts as gateway, so the HSM sync/backup traffic is routed through the VPN interface (qcicat0).

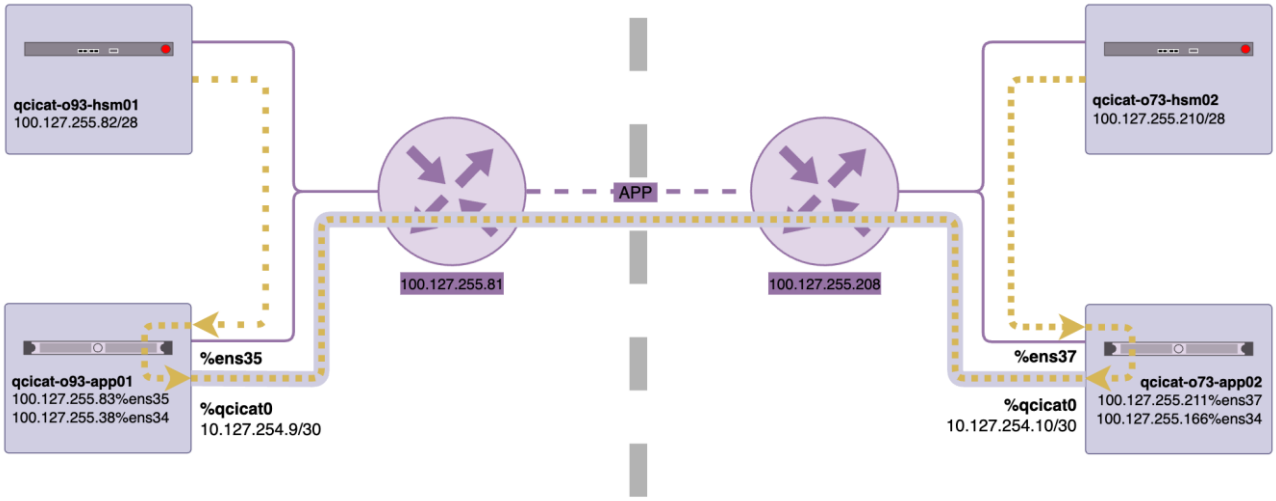


Figure 12 – routing path through the VPN

Additional Source NAT (Network Address Translation) is required on the application server for the proper HMS interconnectivity. Which can be done via **post-up** and **post-down** hook in the WireGuard configuration.

WireGuard configuration snippet of */etc/wireguard/qcicat0.conf* on *qcicat-o93-app01* with the related hooks.

```
## SNAT workaround for onhost lunacm connections to HSM

PostUp = iptables -A FORWARD -i qcicat0 -j ACCEPT; iptables -t nat -A POSTROUTING
! -s 100.127.255.82 -d 100.127.255.210 -j SNAT --to-source 100.127.255.83
```

```
PostDown = iptables -D FORWARD -i qcicat0 -j ACCEPT; iptables -t nat -D
POSTROUTING ! -s 100.127.255.82 -d 100.127.255.210 -j SNAT --to-source
100.127.255.83
```

6.3.5. CLI tools

With the CLI tools it is possible to show the functionality of the solution, such as interactions with the components or the display of the key for demonstration purposes.

However, in a production environment the key must not be shown or stored outside the secure perimeters, and it is important to make sure that all the security measures in place protect the key material from leaking.

Console view of Arnika and WireGuard in Demo Mode where keys are exposed to showcase the principles:

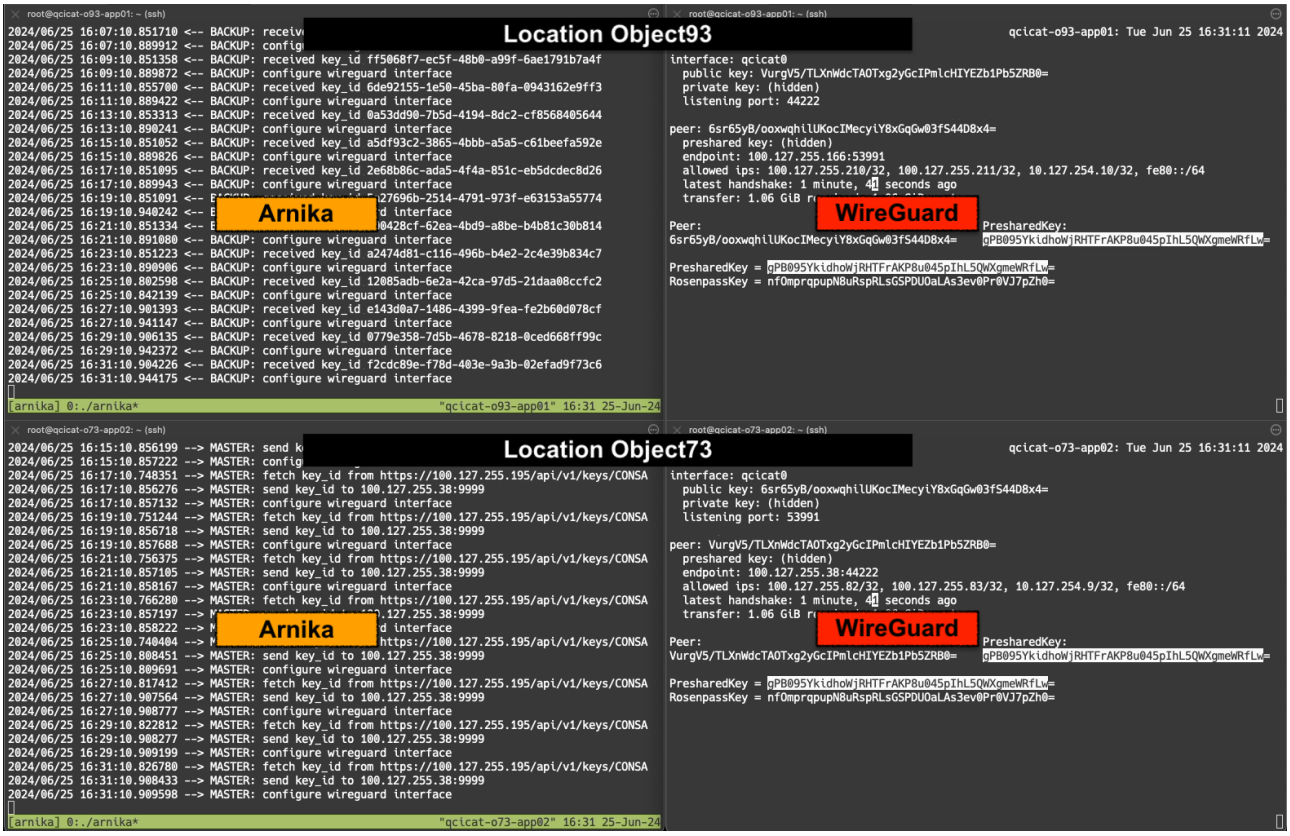


Figure 13 – Arnika interacting with WireGuard

In the left-hand panes, entitled “Arnika”, the key control log is displayed. This log shows the 'key_id' interactions between Arnika instances, as well as the “ETSI014” interaction with the QKD-KMS and the pre-shared key injection into WireGuard. In the right-hand panes, entitled “WireGuard”, an output of a script is presented. This output includes the PQC key from Rosenpass, as well as the final derived pre-shared key and output of the “wg” tool with VPN parameters and metrics.

7. Demo APP

The Demo APP consists of two instances, one instance on each side of the VPN tunnel that both interact with the HSM on its side. There is no direct interaction between the Demo APP and the peer side HSM.

The Demo APP accesses the local HSM via the PKCS#11 API and can perform several operations against the HSM such as:

- List of slots (on HSM) (GetSlotInfo)
- List of tokens (GetTokenInfo)
- List of Public/Private Keys (FindObjects)
- Encrypt message via Public Key (from Token)
- Decrypt message via Private Key (from Token)
- Delete Key (destroyObjects)
- Generate a new (private) keypair (generateKeyPair)

The purpose of the Demo APP is purely to demonstrate that the generated key material, once created on one site will become available on the other site via normal HSM partition replication (sync/backup) mechanisms being forced through the quantum-secure VPN tunnel.

The Demo APP does not interact in any way with the QKD or VPN in this setup.

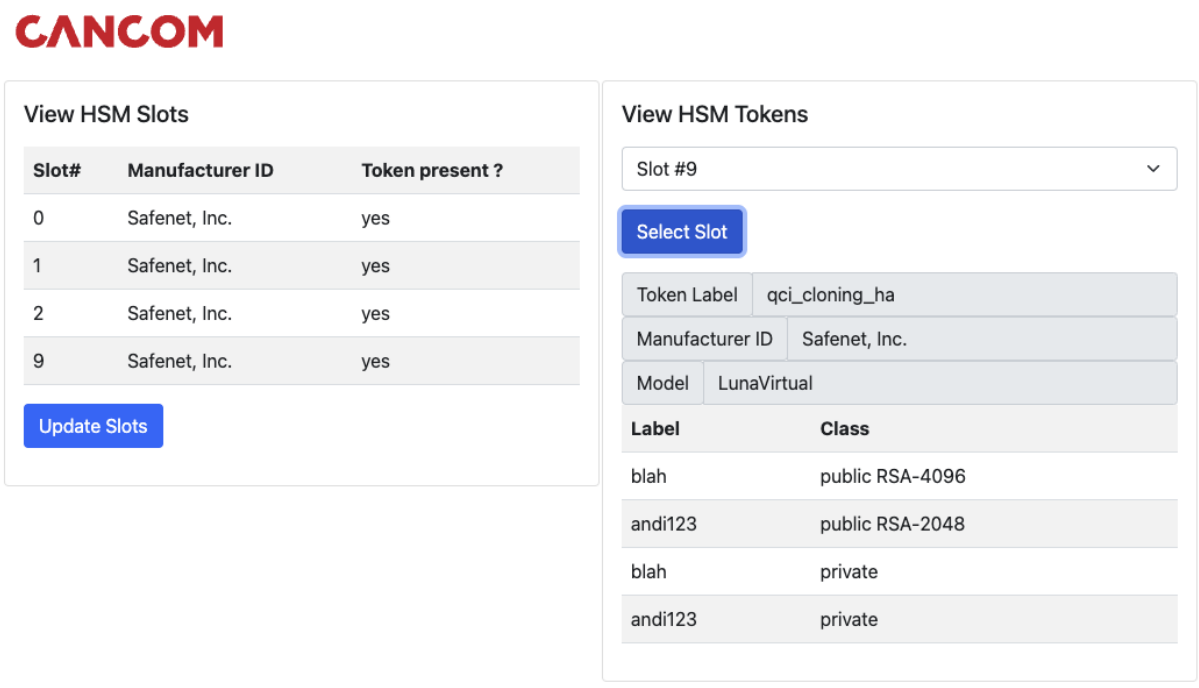


Figure 14 - Demo WebAPP - PKCS#11 interface to HSM.

Encryption operation (Side A):

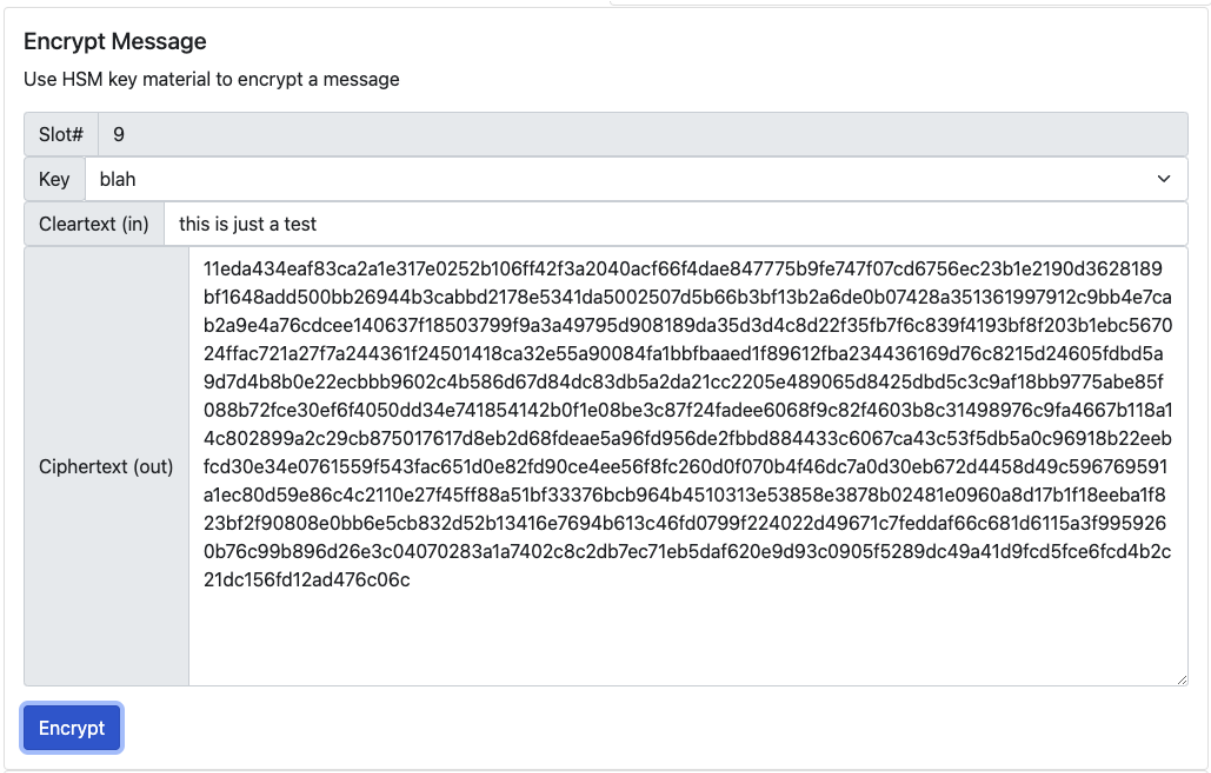


Figure 15 - Demo WebAPP - Site A - encrypting message with HSM key from slot 9.

Decryption operation (side B):

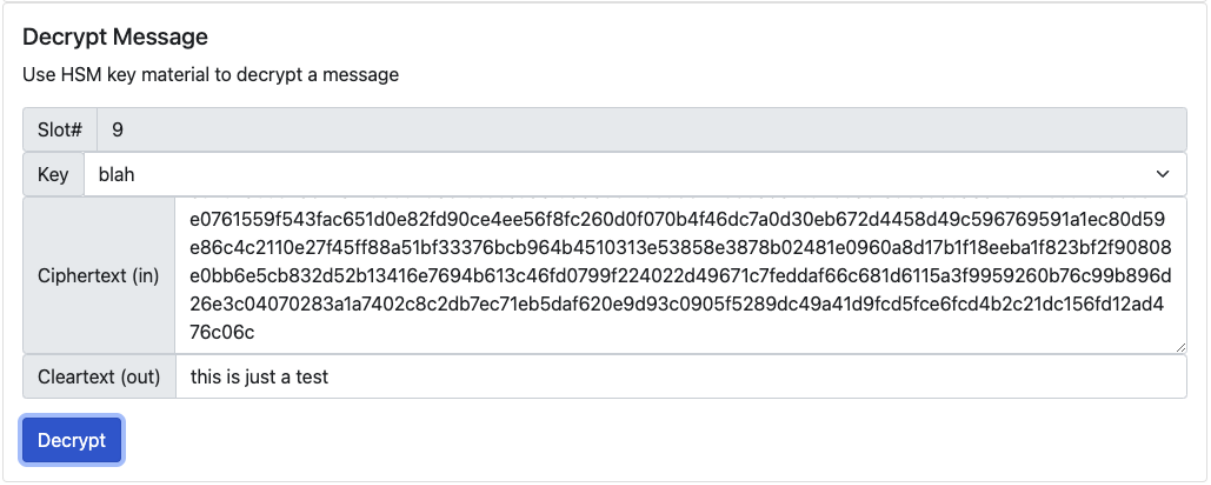


Figure 16 - Demo WebAPP - Site B - decrypting message with HSM key from slot 9.

8. Infrastructure

The whole infrastructure in/to two data centers located in Vienna was built from scratch. Given its direct accessibility via the public internet, stringent access controls were put in place. Incoming access was restricted to a few whitelisted IP addresses, and connections to the platform were only permitted through personal VPNs, ensuring that only authorized personnel could gain entry.

To further enhance security, the firewall feature of the MikroTik was enabled to prevent unauthorized access from the internet. These firewalls acted as a robust barrier, blocking any potential threats and ensuring that the environment remained secure.

The setup included comprehensive security measures, such as VPN access and firewalls, to shield the infrastructure from the public internet and protect it from any unauthorized access attempts. This approach ensured that the infrastructure was not only able to function independently but was also highly secure.

8.1. Data Center

In each data center, the use-case was implemented within a separate, dedicated environment to ensure isolation and security. Both data centers had Rackspace and public subnets ordered and allocated using the same process as for onboarding new external customers. This approach guaranteed that there was no connection or overlap with any existing company infrastructure, maintaining a clear separation.

The data centers comply with several key standards and regulations to ensure security, privacy, and operational efficiency such as **ISO 27001** which specifies requirements for an information security management system (ISMS) and **EN 50600** that provides comprehensive specifications for the planning, construction, and operation of data centers, covering aspects like building construction, power supply, environmental control, and security systems. These standards ensure that data centers operate securely and efficiently, protecting sensitive data and maintaining high levels of service.

The NTT Vienna 1 Data Center holds several security certifications and complies with various regulations. Some of the key certifications include: **ISO 9001**: Quality Management Systems, **ISO 27001**: Information Security Management Systems, **ISO 22301**: Business Continuity Management Systems, **ISO 50001**: Energy Management Systems, **ISO 45001**: Occupational Health and Safety Management Systems, **PCI DSS**: Payment Card Industry Data Security Standard, **TISAX**: Trusted Information Security Assessment Exchange, **EN 50600**: Data Center Facilities and Infrastructure Standards, **ISAE 3402 Type 2** and **ISAE 3000 Type 2**: Assurance Reports on Controls.

These certifications ensure that the data center meets high standards for security, quality, and operational excellence. **References:** Compliance and Certification - NTT, Vienna 1 Data Center, EMEA Vienna 1 Data Center

The two data centers are located approximately 2.5 kilometers apart by road, with the dark fiber length measuring around 3.8 kilometers. Please refer to the map of Vienna below for their location.

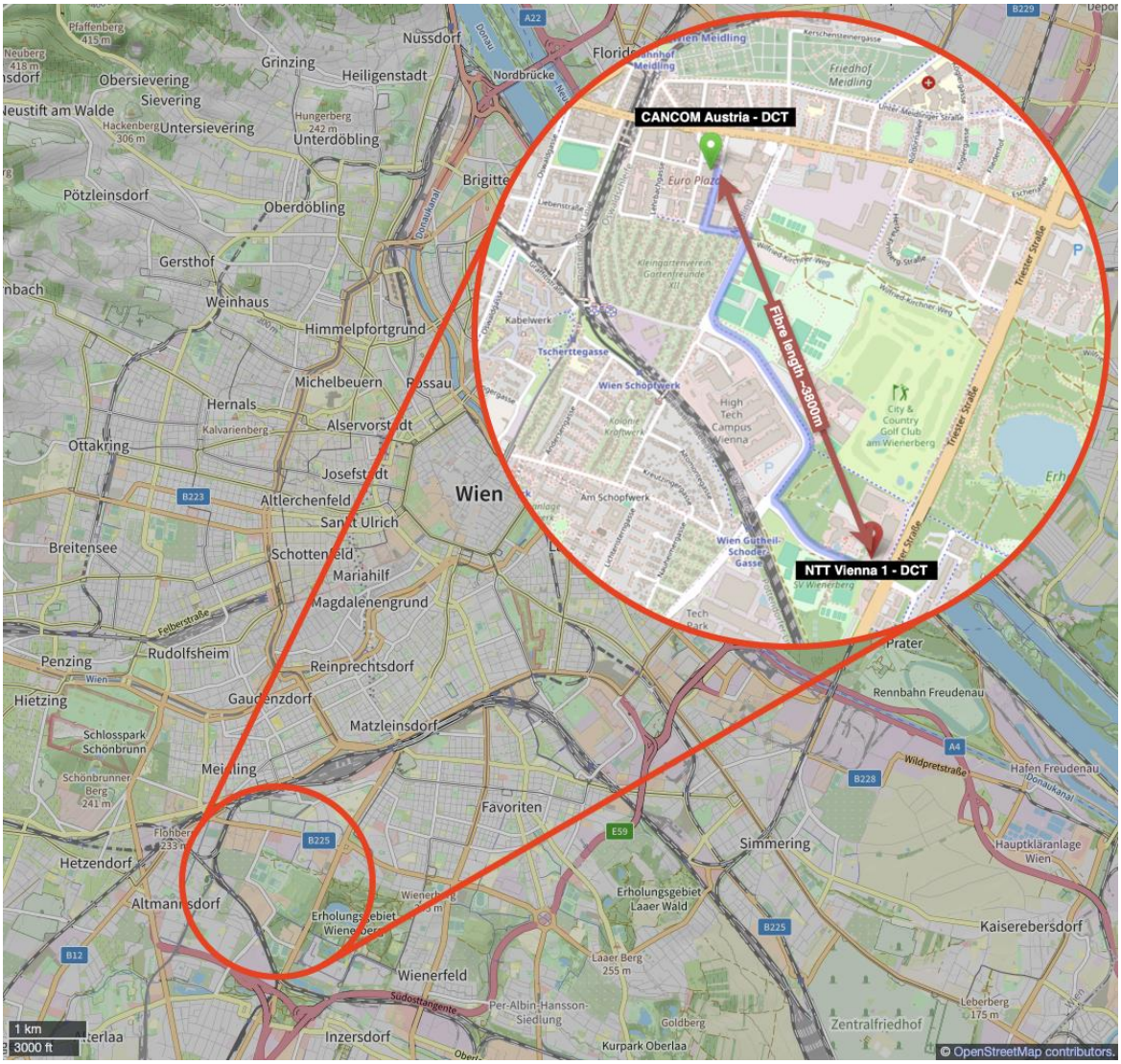
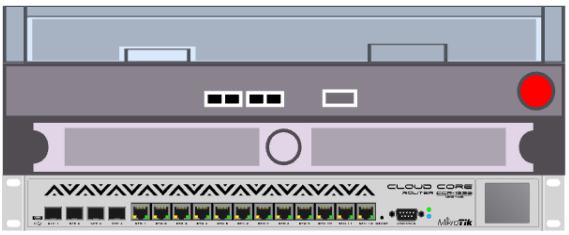


Figure 17- Location of data centers CANCOM Austria and NTT Vienna, 3.8 kilometers apart.

Due to the compact design of the IDQ OKD devices and the other equipment requiring only one rack unit each, the total infrastructure needed a total of just four rack units, allowing for straightforward rental of a quarter rack.



04 QKD IDQ Cerberis XG
03 Thales Luna Network HSM 7 – A700
02 HPE ProLiant DL160 Gen10
01 Mikrotik CCR2116-12G-4S+

Figure 18- Rack plan.

Figure 19- Rack front in DCT NTT Vienna 1.

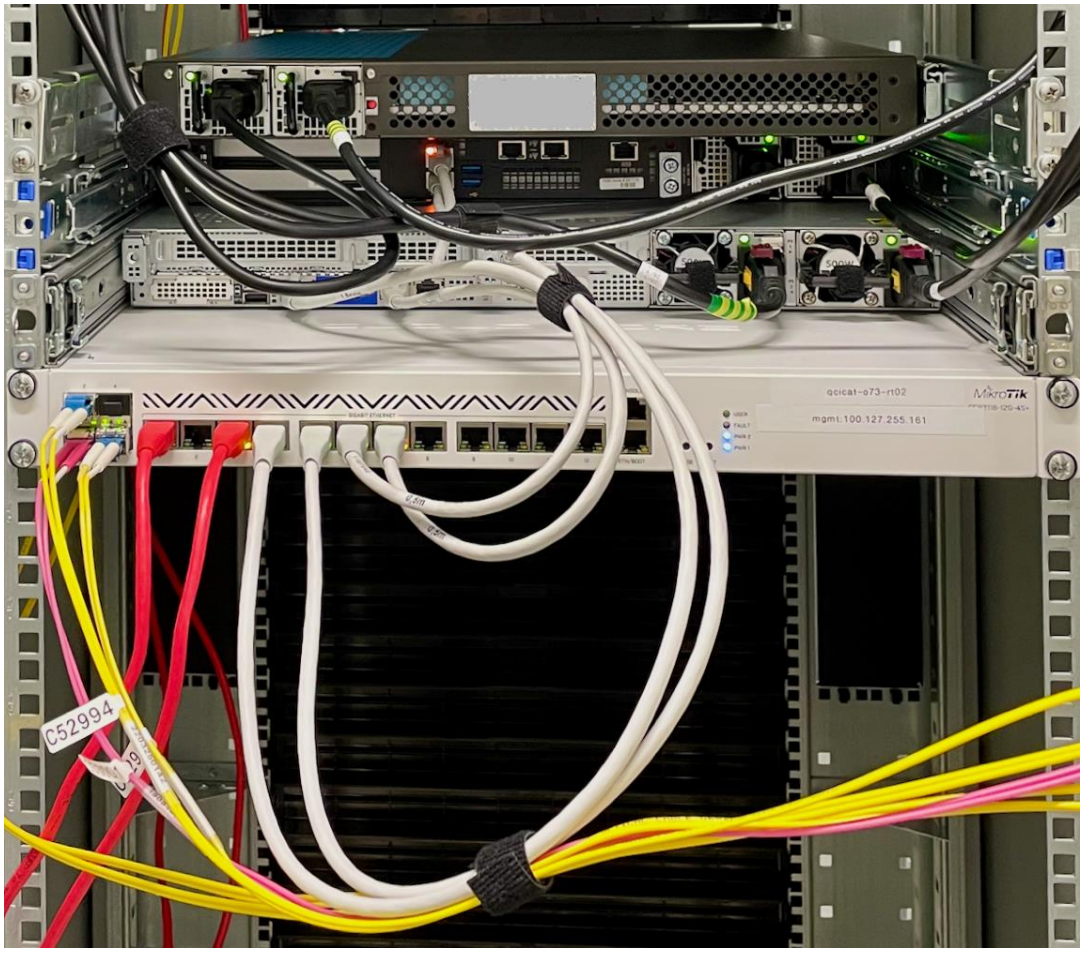


Figure 20- Rack back in DCT NTT Vienna 1.

8.2. Fibers

For the upstream infrastructure, which connects the public internet to the environment, we use multi-mode fiber to provide a 1 Gbit up/downlink.

For the direct crosslink between the two data centers, we utilized four single-mode dark fibers, each spanning approximately 3800 meters. Sometimes there seems to be misconceptions about the term “dark fiber”. This is why it is important to make it very clear in the requirements and clarify that dark fiber means that there are absolutely no active components in between, no attenuators, no DWDM. The photons entering and traveling through the dark fiber must remain intact and emerge at the end of the fiber. This requirement applies to the entire end-to-end connection.

For planning the SFP link budget, we calculated with 4km of fiber optic cable length and a total attenuation including all patch cables of 5dB.

For all the technical details of the fiber characteristic please refer to the OTDR measuring protocol for 1310nm and 1550nm of one of the 4 fibers, shown below in Figure 21, Figure 22 and Figure 23.

Print Date : 13/12/2023 13:23 Fiber Optics : 23.34 File : QCICAT SM_eS-0637001.msor.pdf

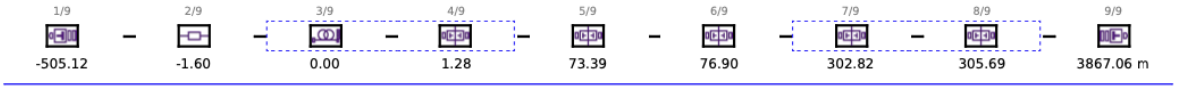
VIAVI

Cable Id :
Location A : Europlaza BTC DC F01
Job ID : QCICAT
Fiber Id/Number : SM_eS-0637 1
Location B : NTT CMMRB
Technician Id : Ramschak

Summary

Filename	Laser nm	Link Loss dB	Link Orl dB	Fiber End m	Direction	Avg Loss dB/Km	EventAlarms
QCICAT SM_eS-0637001.msor	1310	3.459	38.23	3867.43	Europlaza BTC DC F01 -> NTT CMMRB	0.894	11
	1550	3.162	41.65	3867.06	Europlaza BTC DC F01 -> NTT CMMRB	0.818	11

Alarms
Thresholds None



Page Number : 1

Figure 21- OTDR fiber measuring protocol – page 1/3

Print Date : 13/12/2023 13:23 Fiber Optics : 23.34 File : QCICAT SM_eS-0637001.msor.pdf

VIAVI

Cable Id :
Location A : Europlaza BTC DC F01
Job ID : QCICAT

Fiber Id/Number : SM_eS-0637 1
Location B : NTT CMMRB
Technician Id : Ramschak

MTS 4000 V2 (S/N EBAH11735) 4126 B (S/N 03923) Calibration date : 22/02/2023 Date : 13/12/2023 13:22 (UTC+2)

Test Setup

OTDR	Laser	Pw	Range	Acq Time	IOR	Scatter Coefficient
EXPERT	1310nm	5ns	5km	30.0s	1.46750(G652 G657)	-79.0 dB

Alarms

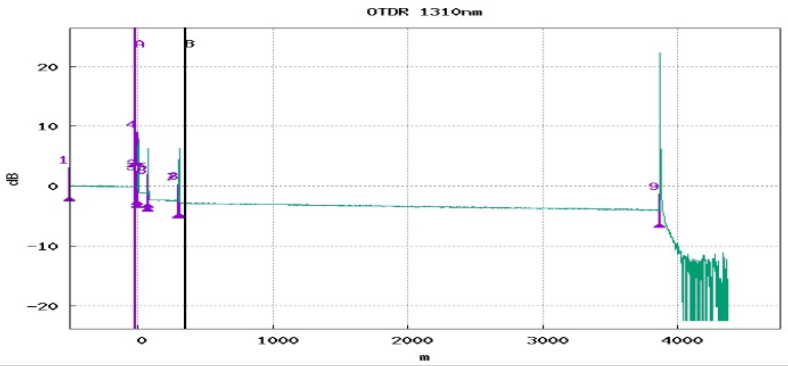
Thresholds	None

Summary

Filename	Laser nm	Link Loss dB	Link Orl dB	Fiber End m	Direction	Avg Loss dB/Km	EventAlarms
QCICAT SM_eS-0637001.msor	1310	3.459	38.23	3867.43	Europlaza BTC DC F01 -> NTT CMMRB	0.894	11

A : -23.48m -0.137 dB
B : 353.95m -2.893 dB

A-B : 377.44m 7.302 dB/Km **2.756 dB**



Event		Distance	Loss	Reflect.	Section Att.	Section	T. Loss
		m	dB	dB	dB	m	dB
1		-504.97	~	~ -75.27	0.000	0.00	
2		-2.55	0.412		0.175	502.42	
		0.00	0.538	-53.48	0.000	2.55	
3		0.00		-55.77			
4		1.28		-53.48			
5		73.74	0.265	-55.34	0.001	72.46	0.539
6		76.93	0.994	-54.65	0.000	3.19	0.804
		302.92	0.453	-52.31	0.079	225.99	1.877
7		302.92		-56.07			
8		305.79		-52.31			
9		3867.43		>-17.19	1.129	3561.63	3.459

Page Number : 2

Figure 22- OTDR fiber measuring protocol – page 2/3

Print Date : 13/12/2023 13:23 Fiber Optics : 23.34 File : QCICAT SM_eS-0637001.msor.pdf

VIAVI

Cable Id :
Location A : Europlaza BTC DC F01
Job ID : QCICAT

Fiber Id/Number : SM_eS-0637 1
Location B : NTT CMMRB
Technician Id : Ramschak

MTS 4000 V2 (S/N EBAH11735) 4126 B (S/N 03923) Calibration date : 22/02/2023 Date : 13/12/2023 13:22 (UTC+2)

Test Setup

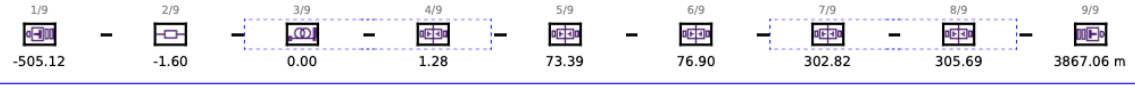
OTDR	Laser	Pw	Range	Acq Time	IOR	Scatter Coefficient
EXPERT	1550nm	5ns	5km	30.0s	1.46800(G652 G657)	-81.0 dB

Alarms

Thresholds: None

Summary

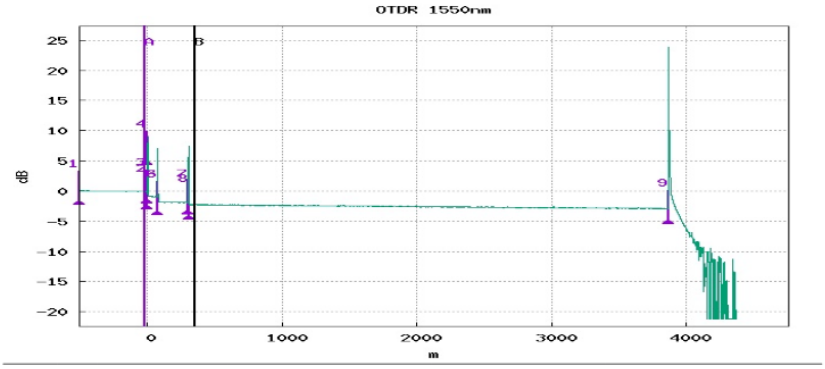
Filename	Laser nm	Link Loss dB	Link Orl dB	Fiber End m	Direction	Avg Loss dB/Km	EventAlarms
QCICAT SM_eS-0637001.msor	1550	3.162	41.65	3867.06	Europlaza BTC DC F01 -> NTT CMMRB	0.818	11



A : -23.48m -0.059 dB
B : 353.95m -2.267 dB

A-B : 377.44m 5.849 dB/Km

2.208 dB



Event		Distance	Loss	Reflect.	Section Att.	Section	T. Loss
		m	dB	dB	dB	m	dB
1		-505.12	~	~ -71.42	0.000	0.00	
2		-1.60	-0.310		0.101	503.53	
		0.00	1.151	-53.49	0.000	1.60	
3		0.00		-56.79			
4		1.28		-53.49			
5		73.39	0.040	-55.97	0.056	72.11	1.207
6		76.90	0.845	-57.97	0.000	3.51	1.246
		302.82	0.366	-53.02	0.054	225.92	2.146
7		302.82		-56.82			
8		305.69		-53.02			
9		3867.06		>-18.22	0.650	3561.38	3.162

Figure 23- OTDR fiber measuring protocol – page 3/3

9. Network

To establish a secure network environment for the use-case, we applied a **Network Development Process** which offers a systematic framework for designing, constructing, and managing network infrastructure, transitioning from functional architecture to practical implementation. This process includes following stages:

The **Feedback and Review Process** highlights the continuous improvement and risk mitigation through feedback, ensuring open communication and adjustments.

Transition Stages and Definitions detail the progression from functional architecture to network design and connectivity matrix.

Network Architecture focuses on transitioning to a network-centric approach while maintaining security through **Network Security Zones** and the separation of critical components.

Network Security Zones describe the discrete segments within the network, each with specific security controls, emphasizing the principle of "least privilege" to minimize the impact of security breaches.

Network Design provides a detailed plan for implementing the network, including strategies for VLAN and IPv4 subnet demarcation.

The **Network Connectivity Matrix** represents connections between network nodes.

Finally, **Network Compatibility and Requirements** list the requirements for ensuring compatibility and proper network function, such as separate network interfaces for different functions and VLAN support.

9.1. Network Development Process

The Network Development process is a step-by-step plan for designing, building, and managing network infrastructure.

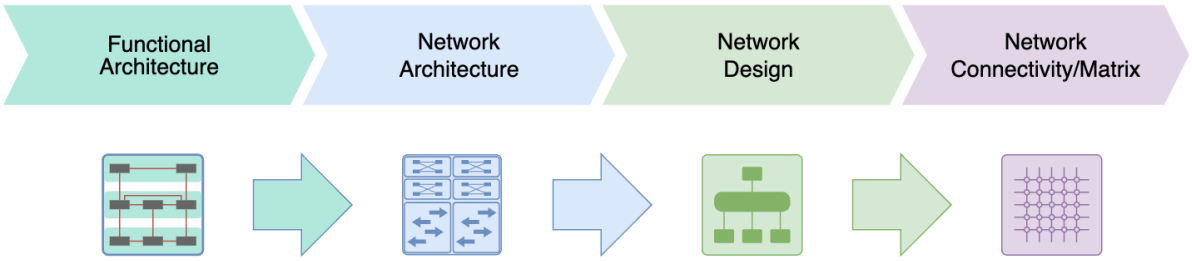


Figure 24- transitioning through various architecture and design stages

This process involves a methodical progression from the functional architecture to the real network infrastructure. It includes a series of steps that bridge the gap between theoretical design and practical implementation.

A structured implementation by this process ensures a systematic approach to develop a complex network and mitigate the risk of oversights and errors.

9.1.1. Feedback and Review process

Transitioning from an architecture to a low-level design can reveal pitfalls that may have originated in earlier stages. A feedback and review process is crucial to identify and address these pitfalls effectively.

Establishing a feedback and review process ensures continuous improvement, mitigates risks, and maintains the integrity and performance of the network. This results in a network architecture and design that remains effective and resilient and is capable of adapting to challenges and evolving needs over time.

It is important to facilitate open communication between the parties involved at every stage. The communication of changes and recognized pitfalls to prior stages enables the elimination of misunderstandings and the correction of possible architectural or design flaws.

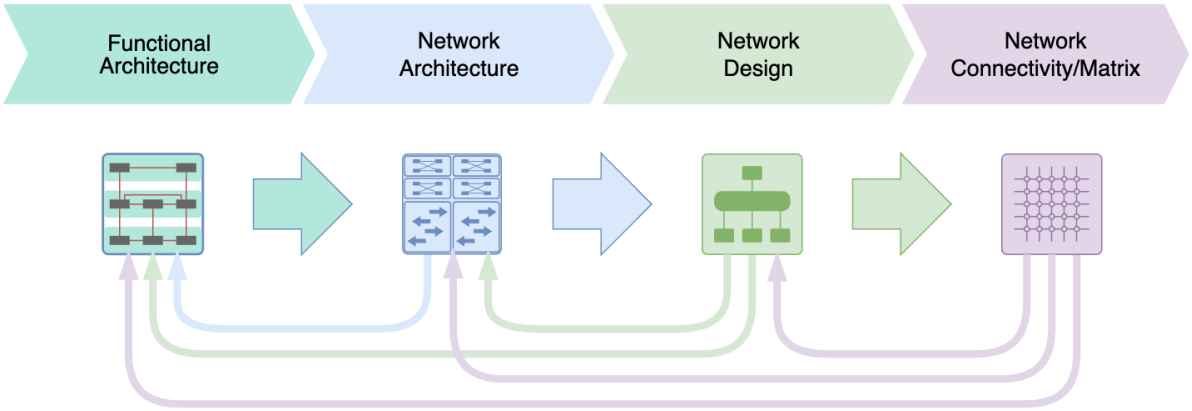


Figure 25 – Feedback and Review process

9.1.2. Transition Stages and definitions

Functional architecture focuses on defining the functions, processes, and responsibilities within the infrastructure. It identifies the services or tasks that each the functions needs to perform and how these interact with each other.

- Focus: Functions and processes.
- Purpose: Define the tasks and services the system needs to perform.

The **Network architecture** focuses on the layout and structure of the network. It defines how the network is organized and how various network components are interconnected. It does not include specific implementation details.

- Focus: Network structure and layout.
- Purpose: Organize the network components and data flow.

Network design translates the architecture into a concrete implementation plan of the network including hardware specifications (e.g., router models, switch configurations), IP addressing and subnetting schemes, cabling plans. Network design ensures that the theoretical framework provided by the network architecture is practically and efficiently implemented.

- Focus: Implementation specifics.
- Purpose: Construct a detailed blueprint for building the network.

Transition to **Connectivity Matrix** outlines the detailed connections between devices. The objective is to provide a clear, concise reference for all network connections, facilitating implementation and troubleshooting.

- Focus: Connections and communication.
- Purpose: Construct a detailed map of device interconnections and permissions.

9.2. Functional Architecture

The following functional architecture, provided by AIT, served as the basis. From this robust foundation, we developed the network architecture and design. Refer to the excellent work described in WP3 “Functional Architecture for the AustroQCI”. The AustroQCI architectural model is based on ITU-T Y.3802 and ETSI QKD 018 with interfaces currently available from ETSI QKD 014/015 and 018.

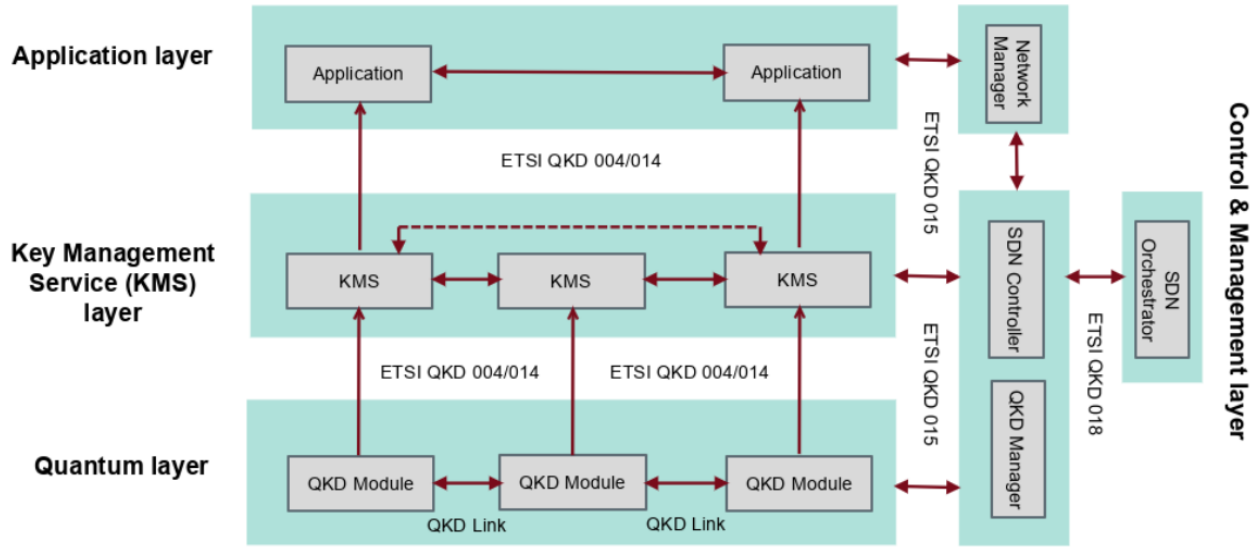


Figure 26 – The AustroQCI architectural model used as a base of the network model.

9.3. Network Architecture

The objective of the network architecture is to move from a functional architectural perspective to a network-centric approach. This must be achieved in a way that ensures that the functional design based on the given QCI-CAT functional architecture remains intact and that no limitations are introduced.

The network architecture is also defined with a focus on network separation including a concept of **Network Security Zones**, which is a concept for the protection of the infrastructure and all involved systems. This approach provides the necessary assurance of integrity, confidentiality, authenticity and availability.

In a QCI infrastructure, it is recommended that the most critical components are physically separated. This means that dedicated and separate network components including switches, routers and firewalls must be employed and isolated from the remainder of the infrastructure.

It is also essential that management and access control functions adhere to the recommended FCAPS framework.

The **network architecture** is fully compatible with the given functional architecture. However, a significant difference is identified in the KMS layer. It was necessary to divide the KMS layer in the network architecture into two distinct layers: a "**Key Distribution**" and a "**Key Management**" layer. The reason for this is to ensure a strict and secure network separation for “Key Management” interactions (eg. KMS-to-KMS) and a client-to-KMS communication. A client/application communication is solely permitted to request keys from KMS via ETSI014. This security-by-design

approach aims to guarantee, that no access is possible from the upper application layer into the “Key management” layer, as any interference with any “Key Management” operations KMS-to-KMS must be avoided in any way.

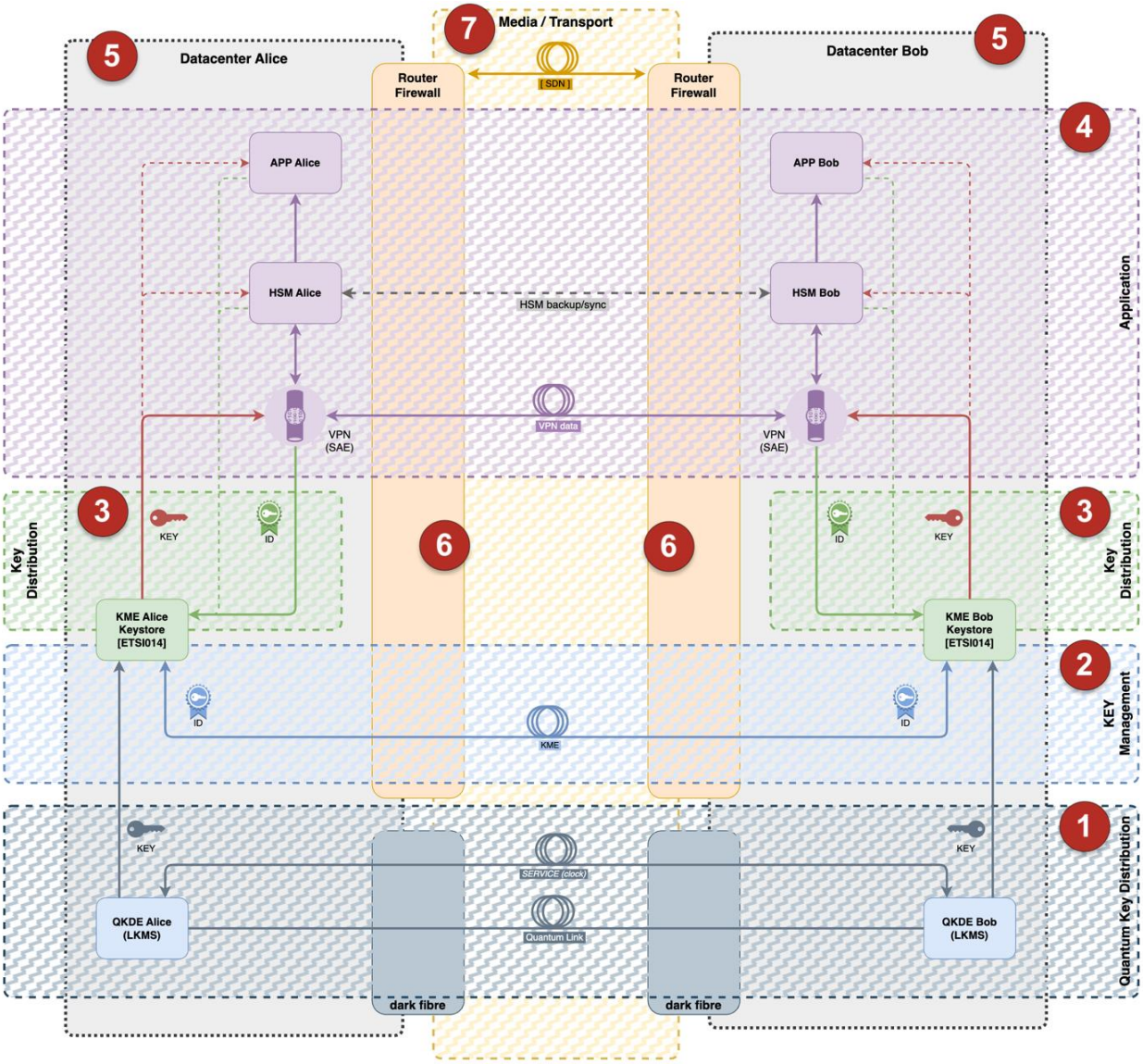


Figure 27 – Network Architecture with (1) Quantum Layer, (2) Key Management Layer, (3) Key Distribution layer, (4) Application Layer, (5) Data Center environment, (6) Firewall and Routing boundary and (7) Transport layer.

1. **Quantum Layer**: This layer encompasses the fundamental physics of QKD. It involves secure key generation based on quantum mechanics principles. Additionally, a separate dark fiber is utilized for a service channel that facilitates proprietary post-processing and time synchronization, ensuring the integrity and accuracy of the key generation process.
2. **Key Management Layer**: Via this layer, the KMS receives key material from the QKD. It also facilitates communication with the KMS on the other site to manage cryptographic keys securely.

3. **Key Distribution Layer**: This layer is responsible for the secure distribution of cryptographic keys to various endpoints and applications via ETSI014 REST interface.
4. **Application Layer**: This layer encompasses the various cryptographic applications and services that utilize the cryptographic keys provided by the KMS.
5. **Site / Location / data center**: This represents the physical locations where the infrastructure and systems are housed, including data centers and other secure facilities.
6. **Firewall and Routing Boundary**: This layer includes security measures such as firewalls and routing protocols that protect the network from unauthorized access and ensure secure data transmission between sites.
7. **Media / Transport**: This visualizes an area beyond a local secure premises, such as external or third-party infrastructure, public networks, service provider's systems any non-owned infrastructure, highlighting the importance of secure transport mechanisms for data traveling through these less secure environments.

9.3.1. Network Security Zones

Network Security Zones are discrete segments within a network, each equipped with security controls (firewalls/routers) to safeguard sensitive data and systems. These zones are established based on the functions performed within them and the level of trust required. The implementation of **Network Security Zones** with increasing levels of network separation can significantly reduce the risk of breaches and improve your overall security posture.

A core principle of **Network Security Zones** is the concept of *"least privilege"*, where systems and users are granted only the necessary permissions to perform their tasks. This minimizes the potential damage from a security breach. Implementing strict access controls between the **Network Security Zones** enables a layered defense strategy, making it more difficult for attackers to move laterally within the network.

In this use case, each network architecture layer was assigned a dedicated network security zone.

9.4. Network Design

The main objective of the network design is to create a comprehensive blueprint that guides the implementation of the network. The network design stage is an important phase in the Network Development Process. It involves detailed planning to ensure that the network meets the requirements defined in the earlier architecture stages.

This includes a strategy for assigning resources for Layer 2 (VLAN) and Layer 3 (IPv4 subnetting).

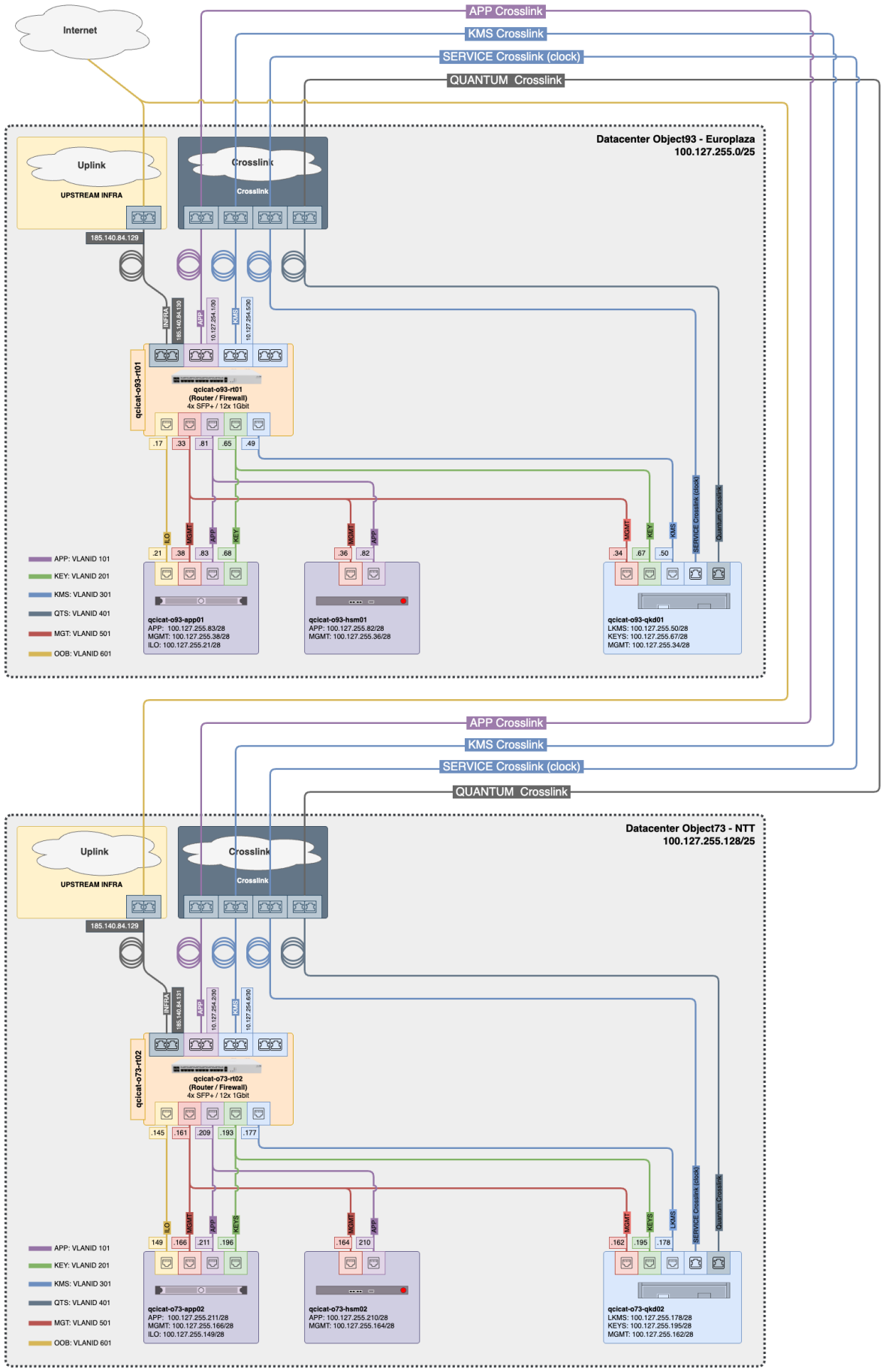


Figure 28 – Full overview of the final network design.

9.4.1. VLAN and IPv4 subnet separation

Each Network Security Zone consists of a dedicated, separated **Layer 2 network** and the associated unique **Layer 3 IPv4 subnet**. The layer 2 network can be separated either physically or via VLAN. In the use-case we focus on separation via VLAN. The firewall and routing functions in the MikroTik CCR2116-12G-4S+ control the traffic between the zones.

9.4.2. Network resources and VLAN assignment

The allocation of network resources, including physical network interface ports and VLAN assignment, is a critical aspect of the network design phase. It is essential for creating a robust, efficient, and secure network infrastructure. The correct configuration guarantees that traffic is properly separated, resources are optimally utilized, and network policies can be effectively enforced.

The table below shows how the network has been separated on the switch via configuring VLAN IDs for the related functions according to the security zone concept. The configuration of the switch is identical on both sites.

Table 1 – Switch port VLAN assignment on switch qcicat-o93-rt01 in site 1-Alice

PATCH Comment	Interface	VLAN 001	VLAN 601	VLAN 501	VLAN 301	VLAN 201	VLAN 101
QKD-MGMT	ether1			native			
QKD-KMS	ether2				native		
QKD-KEY	ether3					native	
HSM-MGMT	ether4			native			
HSM-APP	ether5						native
ESXi-ILO	ether6		native				
ESXi-MGMT-all	ether7			native		trunk	trunk
	ether8						
INFRA-UPLINK	sfp-sfpplus1	native					
APP-CROSSLINK	sfp-sfpplus2						native
QKD-KMS-CROSSLINK	sfp-sfpplus3				native		
QKD-MGMT	sfp-sfpplus4						

9.4.3. IPv4 Addressing Scheme

To ensure scalability and proper segregate of network segments, a subnetting strategy is essential to organize and manage IPv4 addresses.

The IPv4 addresses are unique across the site infrastructure. Any overlap or Network Address Translation (NAT) have been avoided. To limit overlaps and ensure availability the private IPv4 space **100.64.0.0/10** as defined in RFC6598 was used.

IP Address subnet allocation

A /24 subnet has been allocated for the entire use-case which includes 2 sites. For each site a /25 out of the overall /24 has been defined. A separate /24 subnet has been allocated as "linknet" for the purpose of routing traffic/subnets between the sites.

Table 2 – IP Address allocation

	Subnet	Netmask	Usable range
Public	185.140.84.128/29	255.255.255.248	185.140.84.129 - 185.140.84.134
ALL	100.127.255.0/24	255.255.255.0	100.127.255.1 - 100.127.255.254
Object93	100.127.255.0/25	255.255.255.128	100.127.255.1 - 100.127.255.126
Object73	100.127.255.128/25	255.255.255.128	100.127.255.129 - 100.127.255.254
Linknet	10.127.254.0/24	255.255.255.0	10.127.254.1 - 10.127.254.254

IP Address subnet allocation for Object93 - CCA - Wienerberg - Euro plaza

Table 3 – Object93 - CCA - Wienerberg - Euro plaza

Layer 2	Subnet	Netmask	Usable range
INFRA - PVID 1	100.127.255.0/28	255.255.255.240	100.127.255.1 - 100.127.255.14
ILO - VLANID 601	100.127.255.16/28	255.255.255.240	100.127.255.17 - 100.127.255.30
MGMT - VLANID 501	100.127.255.32/28	255.255.255.240	100.127.255.33 - 100.127.255.46
KMS - VLANID 301	100.127.255.48/28	255.255.255.240	100.127.255.49 - 100.127.255.62
KEY - VLANID 201	100.127.255.64/28	255.255.255.240	100.127.255.65 - 100.127.255.78
APP - VLANID 101	100.127.255.80/28	255.255.255.240	100.127.255.81 - 100.127.255.94
Linknet - VLAN 101	10.127.254.0/30	255.255.255.252	10.127.254.1 - 10.127.254.2
WireGuard Clients	100.127.255.8/29	255.255.255.248	100.127.255.9 - 100.127.255.14

IP Address subnet allocation for Object73 - NTT - Vienna 1 - Computerstrasse

Table 4 – Object73 - NTT - Vienna 1 - Computerstrasse

Layer 2	Subnet	Netmask	Usable range
INFRA - PVID 1	100.127.255.128/28	255.255.255.240	100.127.255.129 - 100.127.255.142
ILO - VLANID 601	100.127.255.144/28	255.255.255.240	100.127.255.145 - 100.127.255.158
MGMT - VLANID 501	100.127.255.160/28	255.255.255.240	100.127.255.161 - 100.127.255.174
KMS - VLANID 301	100.127.255.176/28	255.255.255.240	100.127.255.177 - 100.127.255.190
KEY - VLANID 201	100.127.255.192/28	255.255.255.240	100.127.255.193 - 100.127.255.206
APP - VLANID 101	100.127.255.208/28	255.255.255.240	100.127.255.209 - 100.127.255.222
WireGuard Clients	100.127.255.136/29	255.255.255.248	100.127.255.137 - 100.127.255.142

9.5. Network Connectivity matrix

In a managed network, the connectivity matrix represents the connections between various network nodes, such as switches, routers, and end devices. Each element in the matrix indicates whether a direct link exists between any two nodes, with additional information potentially representing the type or quality of the connection.

The connectivity matrix provides a comprehensive overview of the network topology, that can be utilized as input to enforce security policies dynamically, adjusting routes and access permissions based on network conditions and potential threats.

When incorporating firewall functionalities, the connectivity matrix is augmented to reflect security policies. Each matrix element not only indicates a connection but also represents firewall rules that apply to that connection, such as allowed protocols, IP address ranges, and port numbers.

To construct and utilize a connectivity matrix, network engineers first identify all nodes within the infrastructure, then define the connectivity criteria and gather data on the network topology and existing connections. This data is processed to populate the matrix, incorporating firewall rules and policies. The resulting connectivity matrix provides an overview of the network structure and security posture.

Table 5 –Connectivity matrix for ETSI014 and KMS intercommunication.

SRC - Hostname	SRC - IPv4 address	DST - IPv4 Address	DST - Port/Protocol	DST - Service	Comment
qcicat-o93-app01	100.127.255.83	100.127.255.67	443/TCP	REST	ETSI014 connection to KMS
qcicat-o93-qkd01	100.127.255.50	100.127.255.178	9000/TCP - 9001/TCP	KMS	KMS to KMs communication

10. Conclusion

The developed solution, especially the way it uses **QKD** and **PQC** cryptography on Layer 3, has several important advantages:

- Very low key rate -> 1 key per 120 seconds (Rekey-After-Time, Rekey-After-Messages).
- PQC/QKD keys can be injected as pre-shared key at runtime by design.
- No change in existing WireGuard setups.
- An L3-based VPN can operate over any existing, cost-effective, external infrastructure via the internet.
- Rosenpass (PQC) is production ready and already integrated into the commercial VPN service Netbird https://netbird.io/, which supports mesh functionality.
- Unaffected by IPSEC patents like but not limited to "Method of integrating QKD with IPSEC" (US7602919B2, CN101142779A).
- Free and open alternatives to costly proprietary hardware encryptors.
- Ideal for demonstration purposes, as each step can be debugged and made visible.

Appendix A - List of Acronyms

APP … abbreviation for "application"

DNS … Domain Name Service

FCAPS … fault, configuration, accounting, performance, security

HSM ... Hardware Security Module

IDQ … ID Quantique

KMS … Key Management System

NAT … Network Address Translation

PKCS#11 … Public-Key Cryptography Standards #11, also known as Cryptoki, is a standard that defines a platform-independent API for interacting with cryptographic tokens, such as smart cards and HSMs

QBER … Quantum Bit Error Rate

QKD … Quantum Key Distribution

QTS … Quantum Service Network

REST … Representational State Transfer

SDN … Software Defined Network

SSH … Secure Shell

VLAN … Virtual Local Area Networks

VPN … Virtual Private Network

Appendix B – Bibliography

Andreas Neuhold (CANCOM). QCI-CAT: QKD Network Separation. Dok.Nr. 0302400578, 2024. Consortium internal document

Andreas Neuhold (CANCOM). QKD Network Architecture and Design. 2024. Consortium internal document

Christoph Striecks (AIT). Functional Architecture for the AustroQCI. 2024