OPEN
LIGHT
COMM

AIRQKD

AirQKD Key Management Architecture

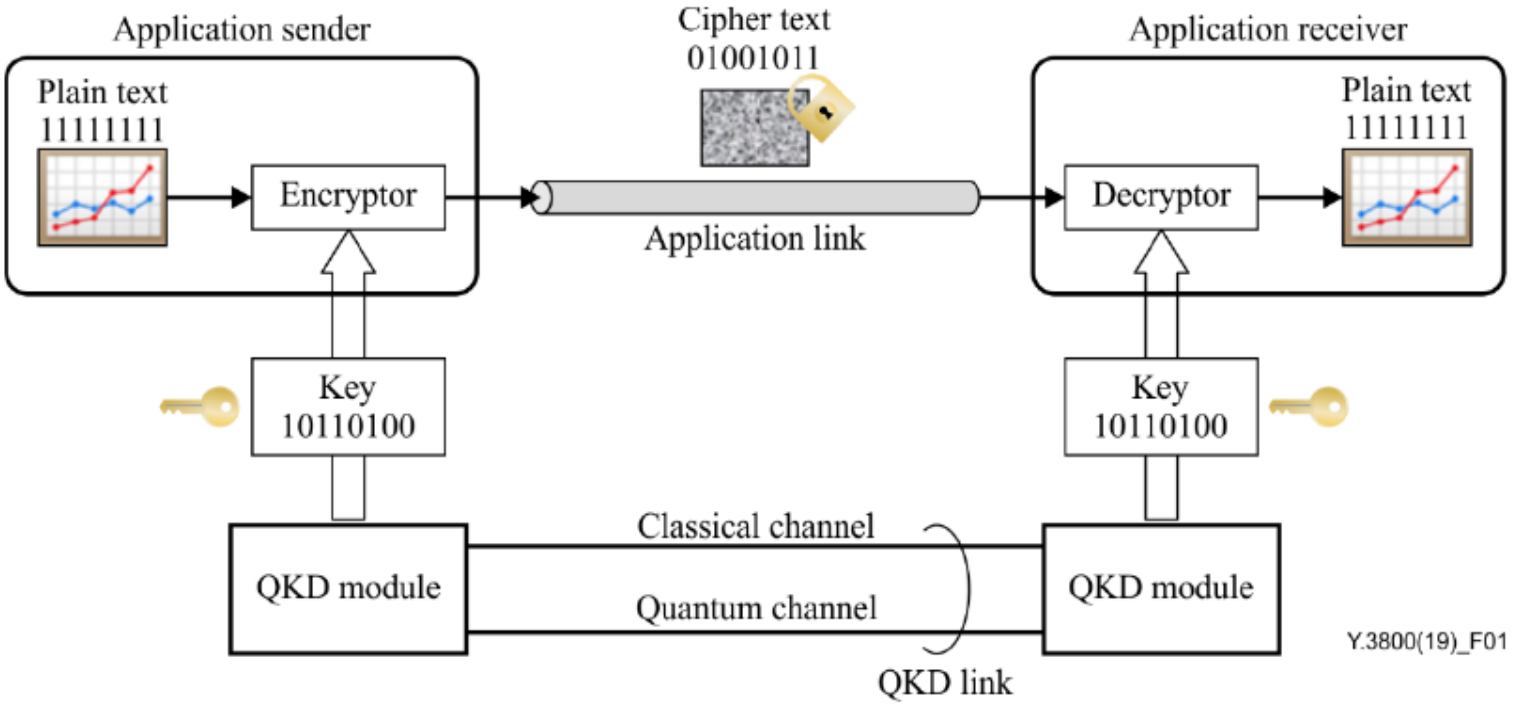
Alexandros Stavdas, Christos Matrakidis, Evangelos Kosmatos
(OpenLightComm)

Overview

- The framework of the ITU-T and ETSI standards
- AirQKD Key Management Architecture
- SDN-based QKD Control/Management Architecture
- SDN-based QKD Orchestration Architecture

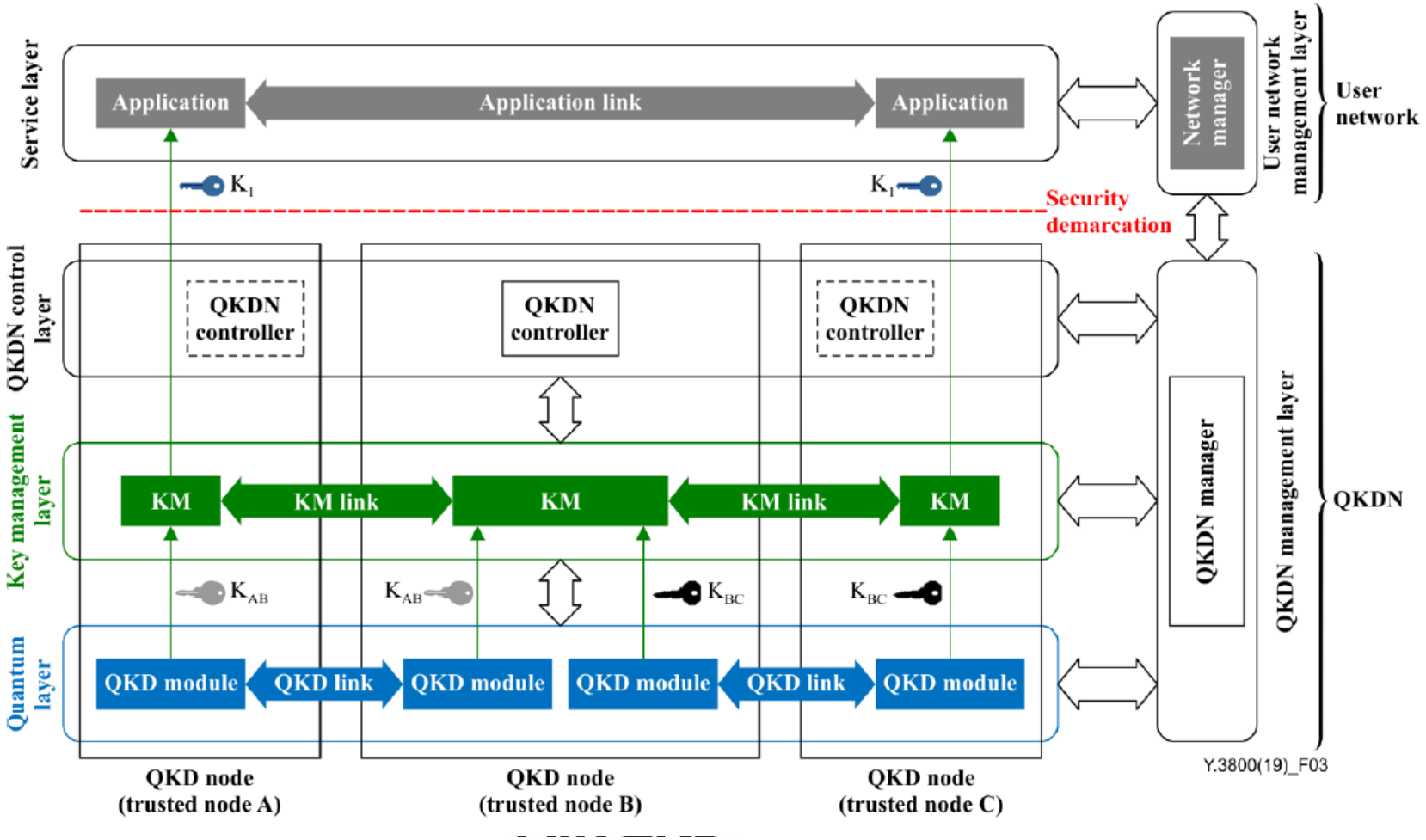
ITU-T & ETSI standardization (I)

- AirQKD Key Management Layer is based on the ITU-T Y.3800 standard

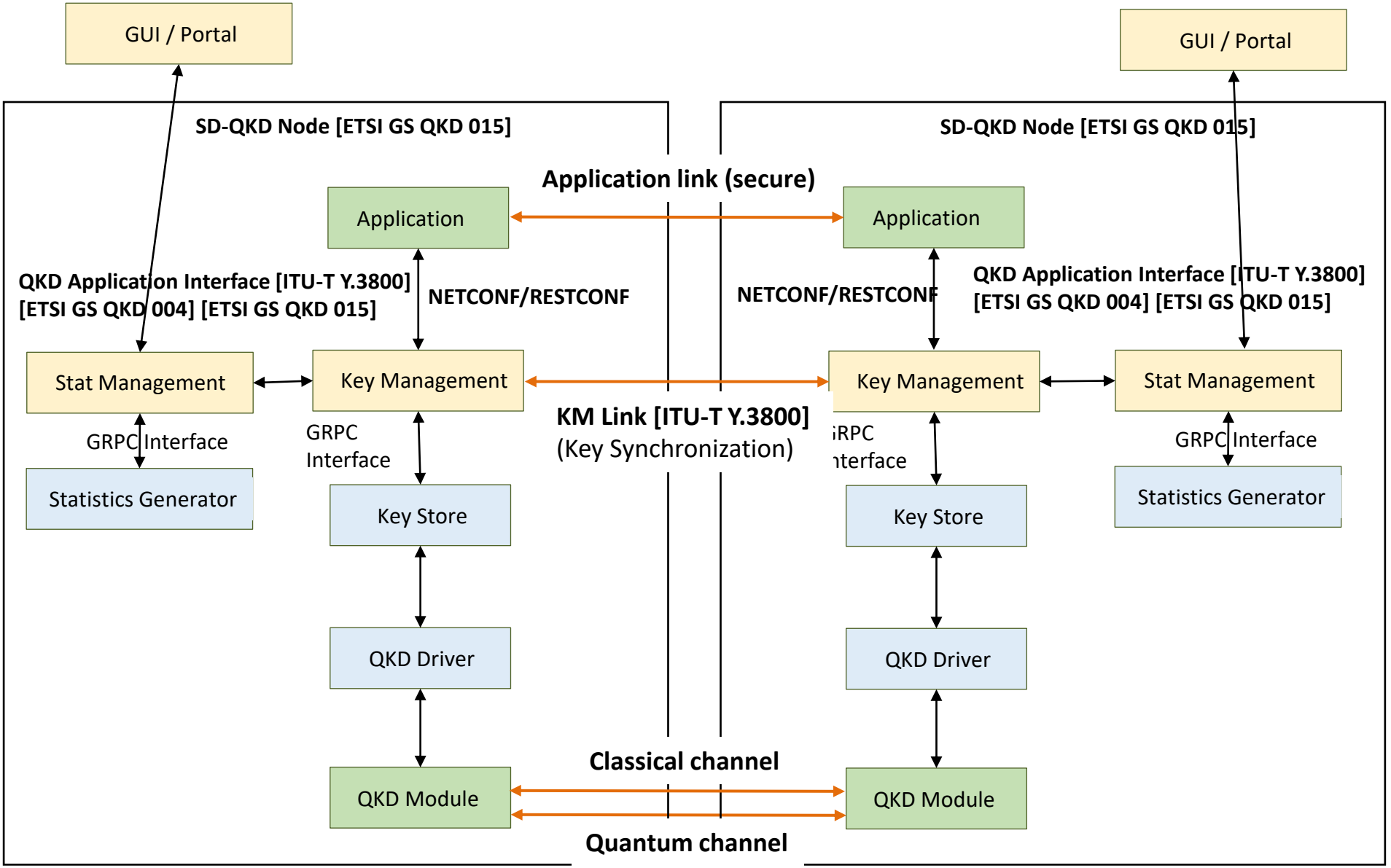


ITU-T & ETSI standardization (II)

- AirQKD Key Management framework follows the layered approach proposed both in ITU-T Y.3800 and ETSI GS QKD standards (ETSI GS QKD 004, ETSI GS QKD 014)



AirQKD's Key Management Layer Architecture



AirQKD's Key Management Module (KMM) -I

- The KMM is central module where the following Key Management functionalities are implemented
  - Key storage
  - Key protection
  - Key identification
  - Key provision to applications on request
  - Key replacement on request
  - Key destruction (based on decided key lifetime)

AirQKD's Key Management Module (KMM) -II

- KMM accepts key requests from the applications and it provides to them the generated keys under the specified QoS.
- KMM exposes an interface (API) to applications based on the **ETSI GS QKD 014, ETSI GS QKD, ITU-T Y.3800 and ETSI GS QKD 015 standards**

QoS parameter	Description
Key_chunk_size	Length of the key buffer, in Bytes, requested by the application
Max_bps	Maximum key rate, in bps, requested by the application
Min_bps	Minimum key rate, in bps, required by the application
Jitter	Maximum expected deviation, in bps, for key delivery
Priority	Priority of the request
Timeout	Time, in msec, after which the call will be aborted, returning an error.
Time to Live (TTL)	Time, in seconds, after which the keys for this KSID shall be erased from the application's dedicated key store, for security reasons.
Metadata mimetype	The mimetype of the metadata to be delivered by the KM on each GET call.

The AirQKD's Key Management Layer

- Applications connects to KMM using HTTPS protocols. At the connection establishment, mutual authentication between Applications and KMS is performed.
- After the mutual authentication, the Application can call API methods on the KMM. The list of API methods provided by KMM are in line with ETSI GS QKD 014

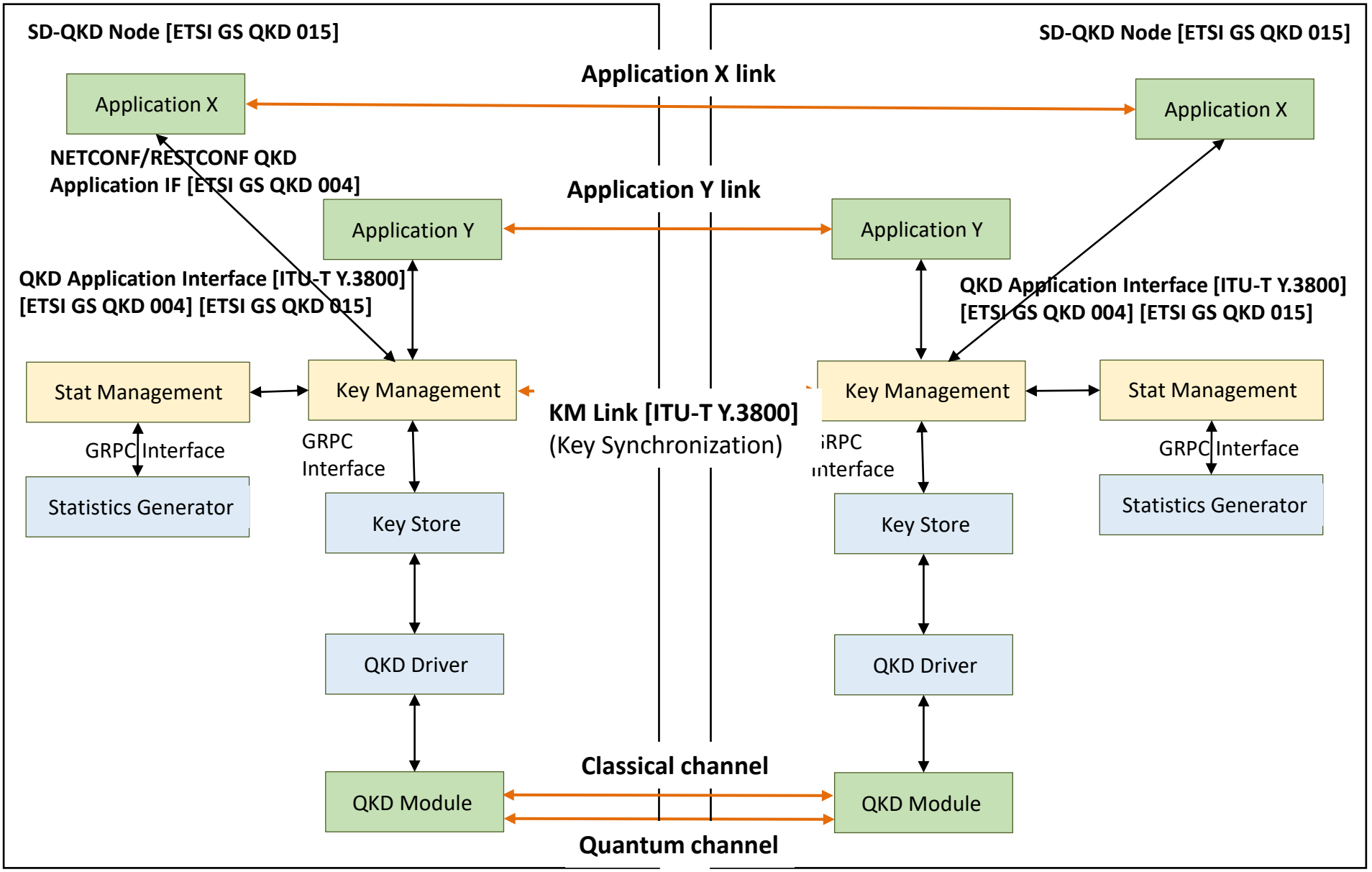
API methods exposed by KMS (described in detail in Sections 5.2, 5.3 and 5.4 of ETSI GS QKD 014)

No.	Method name	URL	Access Method
1	Get status	https://{KME_hostname}/api/v1/keys/{slave_SAE_ID}/status	GET
2	Get key	https://{KME_hostname}/api/v1/keys/{slave_SAE_ID}/enc_keys	POST (or GET)
3	Get key with key IDs	https://{KME_hostname}/api/v1/keys/{master_SAE_ID}/dec_keys	POST (or GET)

The AirQKD's Key Management Layer

- The AirQKD Key Management framework supports monitoring functionalities as well.
- Statistics Generator: poll the QKD driver, collects metrics from it and delivers the metrics toward the Statistic Manager
- Statistic Manager: analyses the collected data and provide results and insights to the end user through the software GUI/Portal.
- Statistics collected includes the following parameters:
  - Keys already used
  - Keys available for use
  - Keys allocated to specific applications
  - Keys expired

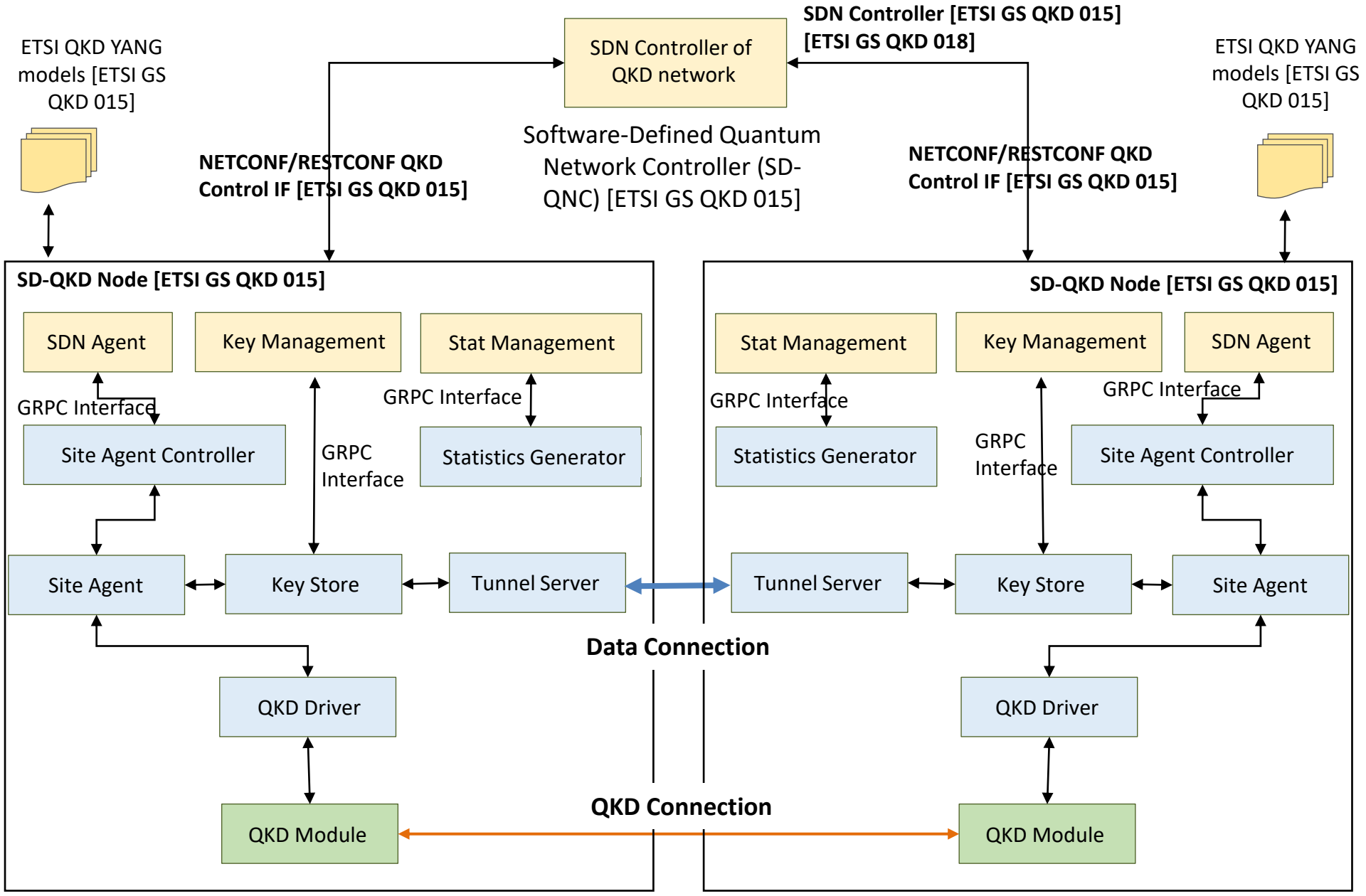
AirQKD Key Management Layer (Multiple applications)



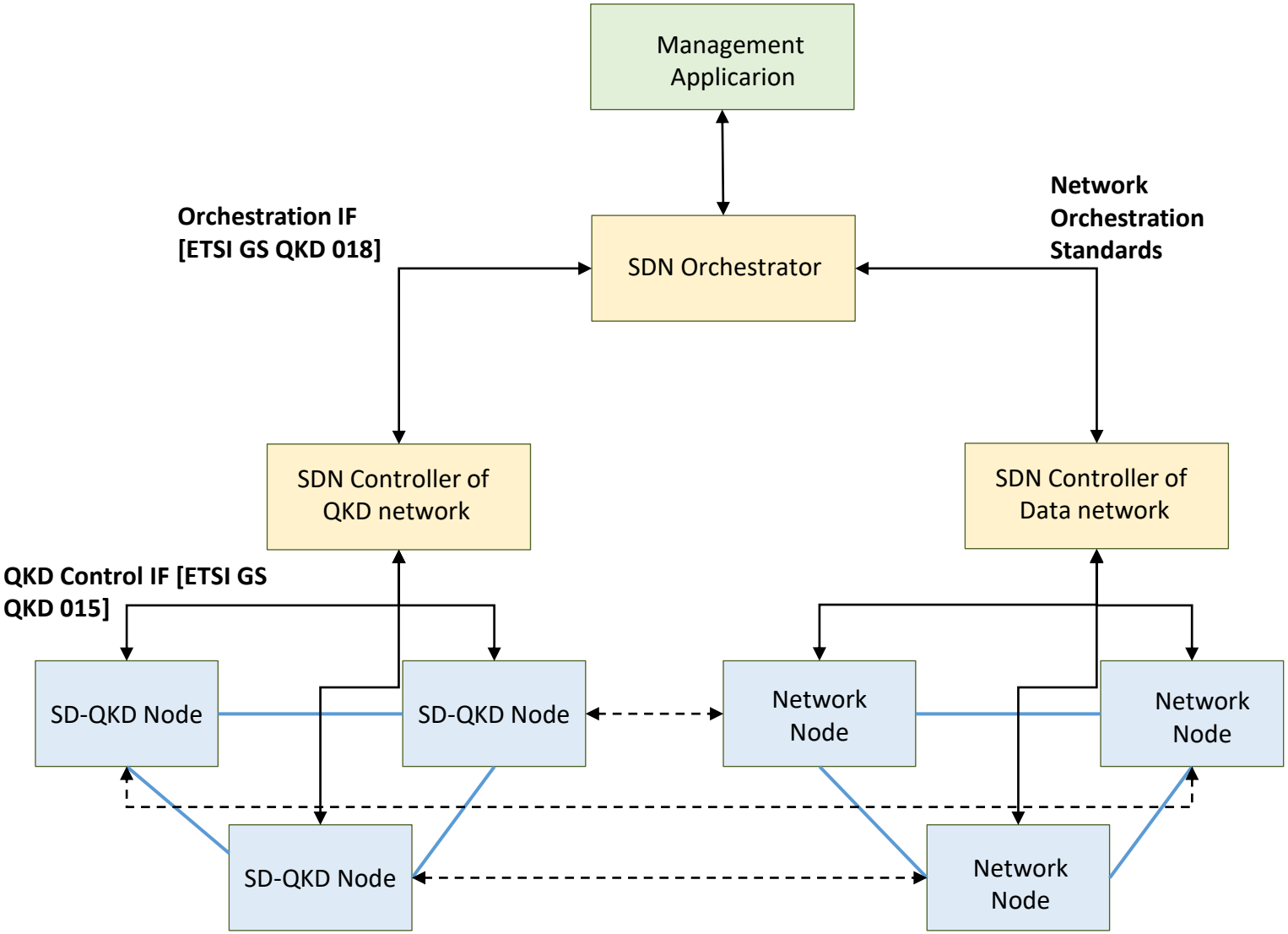
SDN-based QKD Control/Management architecture

- To support BT's procurement, classical communication networks and quantum key generation and transportation need to be jointly orchestrated exploiting SDN principles.
- In AirQKD, the SDN framework is based on the YANG model of the ETSI GS QKD 015.
- The parent SDN Controller of the QKD network exchange messages with the child controllers at every SD-QKD node via the SND by means of protocol the RESTCONF or JSON protocols.
- The SND agent uses an extension of the ETSI GS QKD 014 YANG models and for the purposes of the project it covers the parameters of the QKD applications.

SDN-based QKD Control/Management architecture



The SDN-based Orchestration Architecture



Thank You!!!