



Open Call #2: IMPLEMENT

Annex 2: Technical Annex

Project website	www.pqreact.eu
Opening date:	2 nd April 2025
Closing date:	2 nd June 2025, 17:00h Brussels time
Open call platform:	Application link

* The deadline for submission is as stated in this Guidelines document. Please note that the platform for submission's time depends on the user's configured timezone and may or may not coincide with the correct time (this depends on the user, not the platform for submission). Any discrepancies in system time will not be grounds for deadline extension

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement N° 101119547. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Union's Horizon Europe research and innovation programme. Neither the European Union nor the granting authority can be held responsible for them.



Table of Contents

1. Purpose of the Technical Annex.....	4
1.1 General objectives and key results of PQ-REACT	4
3. Use Case 1: Smart Grid Meters	6
4. Use Case 2: 5G and 6G architectures	8
5. Use Case 3: Context Agility Manager (PQC Benchmarking).....	10
6. Use Case 4: Eclipse - Qrisp for PQC.....	12





Table 1: ACRONYMS

ACRONYMS	
API	Application Programming Interface
CPU	Central Processing Unit
FW	Firmware
HPC	High Performance Computer
IoT	Internet of Things
KPI	Key Performance Indicators
NIST	National Institute of Standards and Technology
PKI	Public key infrastructure
PIC	Participant Identification Code
PQC	Post Quantum Cryptography
QAOA	Quantum Approximate Optimization Algorithm
QC	Quantum computer
QFT	Quantum Fourier transform
QKD	Quantum Key Distribution
UI	User Interface
VPN	Virtual private network





1. Purpose of the Technical Annex

The purpose of this annex is to provide applicants with an understanding of the main technical characteristics of the PQ-REACT framework, architecture and tools, and how the proposals can contribute and execute the validation and demonstration of those tools.

It is important to note that this document constitutes essential and complementary information to the Guidelines for Applicants.

1.1 General objectives and key results of PQ-REACT

The objectives described in this chapter serve as a base to understand what the PQ-REACT project will develop:

- **A Framework:** Project aims to design and build a framework for a faster and smoother transition from classical to post-quantum cryptography for a wide variety of contexts and usage domains. We will create an inventory of cryptographic assets and processes in the cryptographic system to be migrated; A recommendation engine to propose a migration plan, taking under consideration the assets and the various contexts that PQC will be employed.
- **Tools & Architecture:** Project will design architectures and develop innovative approaches and tools that enable cryptographic agility and migration to new cryptographic algorithms and standards in an ongoing way. We will create a tool for optimization of the performance of PQC algorithms on various implementation platforms (from large data centers to IoT devices), taking under consideration delay, memory, CPU and energy consumption restrictions; Strategies for the migration of existing network and telecommunication infrastructures and protocols (e.g. PKI, VPN tunnels, certificate chains, etc.) to post-quantum cryptography; QKD technology as a key exchange method for symmetric cryptography related to 5G networks.

2. General Technical Recommendations

As a fundamental part of the application process, we present this Technical Annex detailing the **essential requirements** and **recommendations** for the successful integration of your Use case projects. Keep in mind that the recommendations in this chapter are to be used with your own discernment as they might not be applicable for your selected Use Case.

This document provides a comprehensive guide on the technical criteria to be considered to ensure the efficient implementation and collaboration within the PQ-REACT platform. By adhering to these requirements, you will not only enhance the functionality of your proposals but also facilitate a seamless integration into this specialized environment.

We urge you to carefully review each section of this annex.





Self-Containment and Accessibility:

- Solutions must be self-contained, encompassing all necessary functionalities.
- Input for the service should be facilitated through an API or configuration file, promoting accessibility and ease of use.

Logging Mechanism for Debugging:

- A logging mechanism is crucial, enabling developers and maintainers to efficiently debug errors and ensure smooth operation.

Hardware Agnosticism:

- Solutions should be executable on off-the-shelf hardware.
- Special hardware components (e.g., GPU, TPM) are not required
- If special hardware is required beyond offered in the technical description of each Use Case, it should be provided by the granted company.

Linux OS Compatibility:

- Solutions must be executable on the Linux operating system, ensuring uniformity across the PQ-REACT platform.

Comprehensive Documentation:

- Use case projects are expected to provide thorough documentation, offering detailed deployment instructions and a user guide for efficient implementation.

Collaboration with Platform Developers:

- Any additional requirements or specific needs beyond this document should be communicated and discussed with PQ-REACT developers during the development phase.

Well-Defined API Specification:

- Solutions must feature a well-defined and documented API (*if applicable for selected Use case*), outlining the required input data and the corresponding results computed by the service.

Headless Execution and UI Integration:

- Solutions should be designed for headless execution on a server without a graphical interface.
- If a user interface is necessary, it must be provided through a web-based UI executable on the PQ-REACT platform or an external tool communicating through the specified API.



The following sections describe each Use Case. The general idea behind each Use Case is described: what challenges the proposal should address, what objectives should it aim towards and what the criteria of completion is. It also describes the desired or suggested implementation as well as methodologies and approach.

3. Use Case 1: Smart Grid Meters

[General description]

Smart Meters are usually provided with limited processing and/or storage capabilities, and also typically deployed in environments with low bandwidth. With this kind of constraints upgrading some of the functionalities protected by Asymmetric Crypto to rely Post-Quantum Crypto algorithms can lead to several problems.

Specifically, the test scenario is focused on the Firmware Update functionality. In some real deployments the FW is protected by means of Elliptic Curve Digital Signatures (e.g.: NIST P-256 Curve) just to address the limitations introduced above, and the Use Case target will be to measure/address the challenges raised by migrating to Post-Quantum Digital Signature Algorithms (or mixed classical/PQC deployments).

[Challenges addressing]

- Key Storage in the limited storage capacity of Smart Meters.
- Firmware Update operations on Smart Meters in low bandwidth environments.
- FW Key Update operations on Smart Meters in low bandwidth environments (i.e.: updating Utility Public Key required to verify the FW signature).
- Firmware Update functionality in the limited processing capacity of Smart Meters.
- Crypto Agility scenarios, where the Post-Quantum Digital Signature Algorithm needs to be updated with some frequency because of PQC Algorithm vulnerabilities.

[Objective of the Use Case/ the project should...]

- Measurement (or estimation) of the impact in specific functionalities due to the migration to Post-Quantum Digital Signature Algorithms protecting the Smart Meter FW. For instance:
 - Comparison of the times required to send the Key Update command in classical/PQC scenarios.
 - Comparison of the times required to send the FW Update command in classical/PQC scenarios.
 - Comparison of the times required by the Smart Meter to verify the received FW in classical/PQC scenarios.
 - Impact in the Smart Meter storage capacity because of higher PQC Keys sizes, also considering security implications (e.g.: inability to store the PQC Keys in secure storage)



Annex 2: Technical Annex

- Possible operational impact because of frequent FW Update operations required by recurring Post-Quantum Digital Signature Algorithm vulnerabilities.
- Etc....
- Proposal of measures to address the found limitations. For instance: the proposal of specific PQC Algorithms depending on specific limitations scenarios:
 - Scenarios where the main limitation is the Smart Meter storage capacity.
 - Scenarios where the main limitation is the Smart Meter's network bandwidth.
 - Scenarios where the main limitation is the Smart Meter's processing capacity.

[Implementation]

The Project must be carried out in real devices (at least the Smart Meters and preferably the Data Concentrators) and in networks that can simulate real environments (for instance a real PRIME/DLMS connection where operational deployment bandwidth and/or noise conditions can be configured).

[Open source]

It is admissible to rely on Open Source projects (e.g.: PQC Crypto implementation, Smart Grid network communication libraries, ...) to complete the Project.

[Completion criteria]

The Project results must include at least:

- Description and capabilities (e.g.: memory, processors, ...) of the devices where the Project is implemented.
- Description and capabilities (e.g.: bandwidth, noise simulation capabilities, ...) of network environment where the Project is carried out.
- Description of the different classical (reference) and PQC Scenarios executed. For instance:
 - Digital Signature Cryptographic algorithms, detailing Key & Signature sizes
 - FW size
 - Network conditions
 - Etc....
- Obtained performance classical/PQC results and comparison analysis.
- When possible, proposed measures to address the found problems.





[Suggested approaches and methodologies]

Whenever feasible, it is recommended:

- To start the Project from an already existing Smart Meter platform and testing environment, in order to focus on the PQC challenged not in the development of the platform and test environment.
- To consider more than one Post-Quantum Digital Signature Algorithm and mixed classical/PQC scenarios.

4. Use Case 2: 5G and 6G architectures

[General description]

Leveraging PQC and QKD hybridization approaches related to 5G and their evolution to 6G architectures. PQ-REACT will investigate their application to significant use cases, especially related to protection mechanisms for the network infrastructure, addressing the security and performance requirements in highly pervasive, critical networked environments.

The use of QKD technology will be offered through the availability of a digital twin of a QKD Network emulation tool (<https://github.com/Networks-it-uc3m/Quditto>). It can also be considered for a demonstration on real Quantum Communication Infrastructure in some of the segments of the EuroQCI-Spain network in Madrid, connected to the 5TONIC testbed (an experimental infrastructure for next-generation network technologies and services) and suitable to extend experiments beyond, through the HellasQCI.

[Challenges addressing]

- Novel integration of standard QKD architectures (ETSI GS QKD 018/20 or ITU-T Y.3818) and PQC to address 5G network connectivity securely.
- Quantum Secure certificate management automation solutions for 5G infrastructures
- Crypto agility solutions for 5G and beyond scenarios
- Performance and impact evaluation on 5G network infrastructure for the PQC use and QKD adoption.

[Objective of the Use Case / the project should...]

The goals of the Use Case involve addressing one or more of the following objectives:

- *Integration of QKD and PQC in 5G networks interconnections*
The objective is to propose solutions and evaluate the performance impact in 5G network connectivity between independent domain providers with the integration of complementary technologies (QKD, classical and PQC), such as hybridization techniques (not limited). The use case will focus on roaming scenarios and associated control (N32), data protocols and certificates and key management between PLMN. Another case can be the use of NPN (Non-





Annex 2: Technical Annex

Public Networks) and its interaction with Public PLMN. Results should be compared with classical existing approaches. The objective includes to measure the impact in:

- IT Resource consumption (CPU, Memory)
- Network resource consumption (bytes per second, packets per second)
- Network performance (delay in secure channel establishment, packet overhead, fragmentation, etc.)
- Techno-economical cost of the alternatives.

Expected outcomes include the identification of the recommended approach supported by previous measurements.

- *Automation on Certificate management transition to Quantum-safe solution*

The objective is to evaluate the performance impact of the management protocols for certificate delivery and use (CMPv3 or Certificate Management Protocol; ACME or Automated Certificate Management Environment) needed by 5G network components and services based on a quantum secure mechanism (PQC and/or QKD). Aspects to evaluate are:

- PQC algorithms selection to address certificate 5G management demands.
- Good practices and guidelines to deploy Certificate management solutions
- Scalability analysis

Expected outcomes include the identification of the recommended approach supported by results.

[Implementation]

The Project will be executed in a virtualized and programable infrastructure environment that will allow participants to deploy their 5G functionalities, specifically those related to roaming connectivity and Certificate management. Radio functionalities are not the focus of this Use Case.

[Open source]

Not mandatory but recommended. Any extension over existing open-source frameworks is welcome to enhance the community and provide guidelines, experiences, and repeatability. Some examples of open-source initiatives aligned with the Use Case are:

- Performance evaluation framework: QUJATA
- 5G networks: Open5Gs, Free5G, UERANSIM
- PKI: OpenSSL
- Crypto libraries: liboqs
- QKD network digital twin: quditto

[Completion criteria]

The completion criteria will require:

- Delivery of developed tools and applications to address the objectives.
 - In case of proprietary solutions, licenses, manual and support during the PQ-REACT project lifetime are required.





- In case of open-source tools, the public access to the same.
- Complete integrated and functional solution covering the previous tools from different participants granted.
- Complete documentation detailing architectural design of the Project software involved, KPI defined, test executed, results and recommendations.
- Solution demonstration and presentation on public event or in a dedicated PQ-REACT consortium event.

[Suggested approaches and methodologies]

Follow-up existing standards and recommendation (GSMA, 3GPP, ETSI, IETF) for PQ transition in the design decision, such as QKD, PQC and hybridization algorithms or performance KPIs to measure.

Adopt open-source solutions, including testing frameworks, crypto libraries and applications, extending public repositories of code.

5. Use Case 3: Context Agility Manager (PQC Benchmarking)

[General description]

Test and validation of innovative services and applications that integrate PQC algorithms across a diverse range of protocols, expanding beyond existing liboqs-based applications such as cURL, Nginx, and HTTP. This involves assessing key performance parameters like CPU usage, memory consumption, power efficiency, and bandwidth utilization in a virtualized infrastructure using a dedicated monitoring tool provided by the project. The initiative will explore PQC-enablers for protocols such as QUIC, SIP, DNSSEC, OAuth2, OpenID Connect, WireGuard, IPSec, and MQTT, ensuring cryptographic agility and interoperability in real-world deployments. The algorithms under evaluation may be NIST-approved or candidates from the NIST Post-Quantum Cryptography Standardization Call.

[Challenges addressing]

The evaluation and validation of PQC-enabled applications introduce several technical and operational challenges, particularly in optimizing resource efficiency while maintaining cryptographic robustness. Key challenges include:

- **Protocol-Specific Integration and Algorithm Diversity:** Each PQC algorithm has unique computational, security, and performance trade-offs. Expanding PQC-enablers to QUIC, SIP, DNSSEC, OAuth2, OpenID, WireGuard, and blockchain-based authentication requires tailored testing methodologies.





Annex 2: Technical Annex

- **Resource Consumption & Efficiency:** Measuring the impact of PQC algorithms on CPU, memory, and power consumption within real-world operational scenarios is crucial, particularly when assessing efficiency in high-performance and constrained environments like IoT and 5G.
- **Performance Trade-offs and Bandwidth Optimization:** Balancing PQC security enhancements with bandwidth efficiency is a key challenge, particularly in latency-sensitive applications such as real-time communications and VPNs.
- **Virtualization Overhead and Scalability:** Differentiating between the computational demands of PQC algorithms and the overhead introduced by virtualized environments is critical to ensuring accurate benchmarking and scalability predictions.
- **Interoperability with Existing Cryptographic Systems:** Ensuring seamless integration of PQC algorithms into widely used communication frameworks without disrupting existing infrastructures remains a core challenge.
- **Reliability of the Monitoring and Testing Framework:** The accuracy of performance metrics, cryptographic agility testing, and energy efficiency benchmarking must be ensured to provide reliable insights for real-world deployment.

[Objective of the Use Case / the project should...]

- **Expand PQC Support Across Protocols:** Develop and validate PQC enablers for a wider range of security and communication protocols, ensuring cryptographic agility and post-quantum security adoption beyond liboqs-based implementations.
- **Assess Scalability and Robustness:** Evaluate how PQC protocols perform under large-scale deployments, high-traffic conditions, and enterprise-grade networking environments.
- **Optimize Energy Efficiency:** Measure and analyze PQC algorithms' power consumption in diverse deployment scenarios, optimizing PQC implementations for energy efficiency in client-server interactions.
- **Ensure Interoperability:** Validate PQC integration with widely used communication protocols and security frameworks, ensuring compatibility with existing infrastructures such as HTTP, WireGuard, and IPSec.
- **Provide Protocol-Specific Performance Testing:** Deliver detailed performance metrics across different PQC-supported protocols, guiding developers in adopting post-quantum security measures effectively.

[Implementation]

The implementation will be carried out across a private virtualized infrastructure provided by NCSRD, allowing experimenters to deploy and assess their PQC-enabled applications. Different hosts will interconnect to evaluate connectivity, security, and resource efficiency across multiple PQC-enforced protocols.

[Open source]





Annex 2: Technical Annex

The underlying testbed will be based on the Open Source framework Qujata. The Qujata project aims to evaluate the performance of supported Post-Quantum Cryptography (PQC) protocols through a testbed that assesses client and server vital signs, including memory and CPU usage, connection time, and download speed. Various Projects applying for the open call can have diverse objectives, reflecting the specific needs and goals of their respective domains.

[Completion criteria]

For the successful validation of PQC-enabled applications, the following criteria must be met:

- **Comprehensive Performance Evaluation:** Accurate measurement and documentation of CPU usage, memory consumption, power efficiency, and bandwidth utilization for each PQC algorithm.
- **Algorithm Compatibility and Interoperability:** Demonstration of seamless integration across security protocols, ensuring compatibility with widely used frameworks such as liboqs.
- **Scalability and Large-Scale Deployment Testing:** Validation of PQC protocols in high-load environments, ensuring robust security under real-world operational conditions.
- **Energy Efficiency and Optimization:** Evaluation of power consumption trends and provision of recommendations for energy-efficient PQC implementation strategies.
- **Reliability of Monitoring Tools:** Ensuring accuracy and consistency in data collection and analysis, minimizing discrepancies in performance benchmarking.

[Suggested approaches and methodologies]

Diverse Algorithm Testing: Implementing and benchmarking a variety of PQC algorithms, including both NIST-approved and candidate proposals.

Hybrid PQC-Classical Deployments: Evaluating mixed deployments of classical cryptographic schemes with PQC algorithms to facilitate smooth migration paths.

Real-World Simulation and Load Testing: Replicating real-world conditions, including latency-sensitive applications, IoT ecosystems, and secure enterprise communications.

6. Use Case 4: Eclipse - Qrisp for PQC

[General description]

Practical validation and assessment of the resilience of new applications/solutions using PQC techniques, exploiting the capabilities of an HPC (High Performance Computer) or QC (Quantum Computer) via a suite of tools (C++/Python/Qrisp). Additional tools like, Quantum Approximate Optimization Algorithm (QAOA), Shor's algorithm, algorithmic primitives (e.g., quantum Fourier





transform (QFT), quantum phase estimation (QPE)) and solving QUBO problem instances with QAOA are available in Qrisp.

The project aims to build a portfolio of open tools for evaluation of the resilience of PQC algorithms and cryptanalytical methods. These tools will enable benchmarking new PQC algorithms and new cryptanalytical methods with quantum computing and high-performance computing resources. The high-level programming framework Eclipse Qrisp, developed by scientists at Fraunhofer FOKUS, enables developers to write quantum code with ease.

[Challenges addressing]

Shor's algorithm, probably the most famous quantum algorithm in cryptanalysis, can efficiently factorize large numbers and thereby break the widely used RSA public key cryptosystem. Therefore, such quantum-insecure algorithms must be replaced by post-quantum cryptography algorithms which are conjectured to be secure against quantum attacks. Typically, the security of such PQC algorithms is based on the hardness of solving a mathematical problem, e.g., learning with errors or multivariate polynomial equations, hash functions. In this Use Case the resilience of relevant PQC algorithms with respect to quantum or hybrid quantum-classical attacks should be validated. It should be tested to see if PQC algorithms are indeed robust against quantum attacks, and under which conditions (e.g. side channel information) they are not secure anymore.

[Objective of the Use Case / the project should...]

A quantum or hybrid quantum-classical attack against a relevant PQC algorithm (i.e., NIST candidates) shall be implemented using open-source tools such as Python and the python-based high-level programming framework Eclipse Qrisp.

- Testing:
 - The attack should be tested against small sized instances of the underlying mathematical problem for the PQC algorithm. Quantum algorithms may be tested on classical devices using the Qrisp simulator.
- Complexity analysis:
 - An analysis of the quantum algorithm in terms of suitable metrics, e.g., qubit count, gate count or circuit depth.
- Scalability analysis:
 - Based on the results of the testing and the complexity analysis, the scalability of the proposed quantum attack should be discussed.

[Implementation]

The implementation shall be carried out using open-source software and libraries such as Python (NumPy, SciPy, etc.), and the python-based framework Eclipse Qrisp for quantum algorithms.

[Open source]





The implementation shall be carried out using open-source software.

[Completion criteria]

To demonstrate the efficiency of the implemented attacker the following items should be considered:

- Generate secret keys of different lengths and try to reconstruct them by means of the implemented quantum or hybrid-quantum attacker. If the PQC-algorithm is based on a learning with error problem (LWE problem), the length of secret key can be increased by increasing the dimension of the underlying LWE problem. A LWE problem consists of a linear system of equations whose matrix and right-hand side are sampled from a probability distribution over a modular ring. The solution of this linear system of equations represents the secret key. Perform the test described in the last item many times. Thereby, the number of successful attacks should be counted for each key length. By this, a success probability of the implemented attacker for a given key can be computed. Represent your results by means of an appropriate data structure or table such that a potential user of the attacked cryptosystem can look up its security level.

[Suggested approaches and methodologies]

The performance of the attackers that have been implemented should be tested by means of a Lattice based cryptosystem that is described in the following NIST report:

National Institute of Standards and Technology, "FIPS 203, Module-Lattice-Based Key-Encapsulation Mechanism Standard," <https://csrc.nist.gov/pubs/fips/203/ipd>, 202

To test the resilience of secret keys with a certain length, the key generator described in the report from above could be reimplemented such that smaller keys can be produced. This is necessary, since many quantum-based attackers can reconstruct only small keys. Furthermore, nowadays quantum hardware also only the handling of small test problems. This is due to the fact that only a few qubits are available and that there is no effective error correction.

Inform yourself about algorithms reducing the computational effort that is required to reconstruct the secret key or the solution of a given LWE problem. Well-known algorithms in this context are the LLL algorithm or the BKZ algorithm. Apart from that information from side-channel attacks could be used.

