



pq-react

Open call 2 - IMPLEMENT
2nd Infoday (Use case 2 & 4)
29/04/2025



Funded by
the European Union

Before We Get Started..

The session will be recorded and the recording forwarded to participants.

Post any questions in the Q&A section and we will answer in the designated Q&A time. If we cannot answer you in the allocated time, please remember you can always reach out to us via the helpdesk email. And if we notice any recurring topics, we'll make sure to include them in the FAQ section on our website.

AGENDA

11:00h - 11:10h Welcome & Overview

Sofia Karamitsiani (NSCRD) & Mikel Apesteguia (Sploro)

11:10h – 11:20h Use Case 2: **5G and 6G architectures**

Victor Hernando (Telefonica)

11:20h – 11:30h Use Case 4: **Eclipse-Qrisp for PQC**

Tobias Köppl (FOKUS Fraunhofer)

11:30h – 11:45h How to apply

Mikel Apesteguia (Sploro)

11:45h – 12:00h Q&A

12:00h – 12:15h Pitching Session



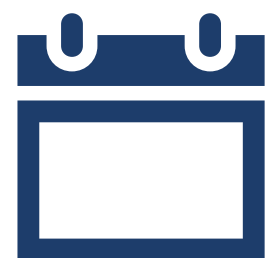
Overview of the project

Post Quantum Cryptography Framework for Energy
Aware Contexts



DURATION

36 months



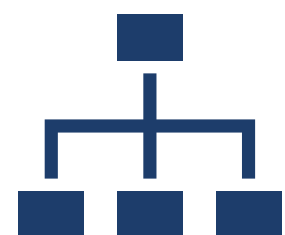
STARTING DATE

01/09/2023



CALL

HORIZON-CL3-2022-CS-01



COORDINATOR

National Center For Scientific Research
"Demokritos" NCSR "D"

Funded by the European Union

Project Details

12 partners across the EU



NATIONAL CENTRE FOR
SCIENTIFIC RESEARCH "DEMOKRITOS"

indra



Fraunhofer



POLITÉCNICA

Sant'Anna
Scuola Universitaria Superiore Pisa



Sploro
disrupting innovation

SmartLex



AGH



uc3m | Universidad
Carlos III
de Madrid



Overview of Open Call 2 - *IMPLEMENT*

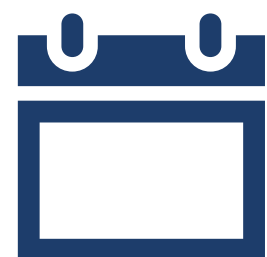


OPEN CALL 2 - IMPLEMENT



DURATION

2 months



DEADLINE FOR APPLICATIONS

27/05/2025 at 17:00h CEST



BUDGET

€650,000 → €162,500 per Use-case (4 winners)



HELPDESK

applications@pqreact.eu

USE CASES:

Use Case 1: Smart Grid Meters

Use Case 2: 5G and 6G architectures

Use Case 3: Context Agility Manager

(PQC Benchmarking)

Use Case 4: Eclipse-Qrisp for PQC

WHY APPLY?

**Funding
Support**



**Contribute to
emerging
standards**



**Visibility and
Recognition**



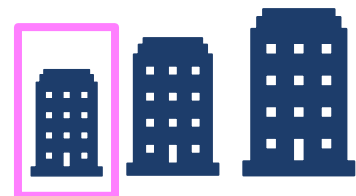
**Be part of
Europe's
quantum
cybersecurity
future**

**Access to
strategic
use-cases**

WHO CAN APPLY?



Consortia of 2 to 3 partners



At least 1 SME



**At least 1 Tech Provider
(SME, University or RTO)**

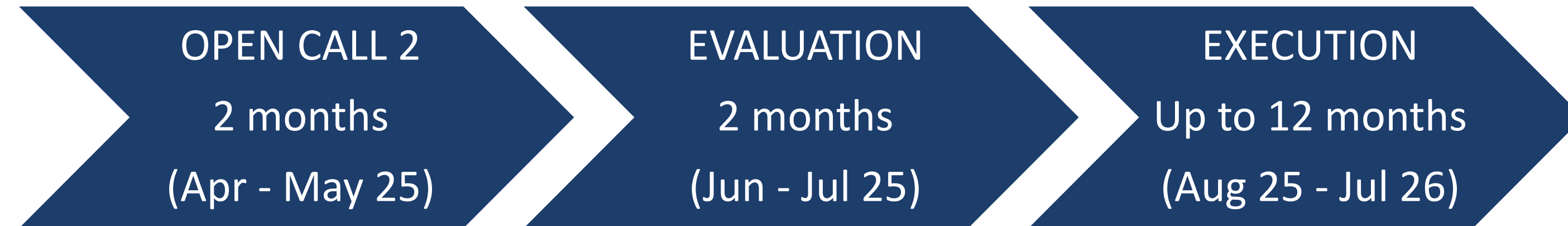


**Eligible entities: SMEs/Startups,
Universities, RTOs, NGOs & Foundations**



**All legally established in
Horizon Europe eligible
countries**

TIMELINE



- Open call duration
- Infodays & Pitching sessions
- Helpdesk

1. Eligibility check
2. Alignment evaluation
3. External evaluation
4. Normalisation of results
5. Final selection
6. Legal validation
7. Sub-grant Agreement signature

- Project implementation



Use case 2: 5G and 6G architectures

Victor Hernando (Telefonica)

Funded by the European Union

General Description

- Leveraging PQC and QKD hybridization approaches related to 5G and beyond architectures
- Investigate their application as protection mechanisms for the network infrastructure, especially related with critical networked environments

Challenges Addressing

Integration of standard
QKD architectures & PQC

Quantum Secure
certificate management
automation


Crypto Agility solutions for
5G scenarios

Performance & impact
evaluation for the PQC use
and QKD adoption


Objectives



Integration of QKD and PQC in 5G networks interconnections

- Roaming scenarios & associated control (N32), data protocols & certificates and key management between PLMN
 - Evaluate the performance impact of the integration of QKD, PQC and hybridization techniques in comparison with classical techniques in **roaming scenarios**.
 - Measure the impact of the integration in:
 - IT Resource consumption
 - Network resource consumption
 - Network performance
 - Techno-economical cost of the alternatives
- 

Automation on Certificate management transition to Quantum-safe solution

- Evaluate the performance impact of the management protocols for certificate delivery and use (CMPv3, ACME) needed by 5G network components and services based on a quantum secure mechanism (PQC/QKD)
 - Aspects to evaluate:
 - PQC algorithms selection to address certificate demands
 - Good practices and guidelines to deploy Certificate management solutions
 - Scalability analysis
- 

Implementation

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

Recommended to use open-source frameworks. Some examples of open-source initiatives aligned with the Use case are:

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

Completion criteria

- Delivery of developed tools and applications to address the objectives
 - Proprietary solutions: licenses, manuals and support during the project lifetime are required
- 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 cosortium event



Use case 4: Eclipse-Qrisp for PQC

Tobias Köppl (FOKUS Fraunhofer)

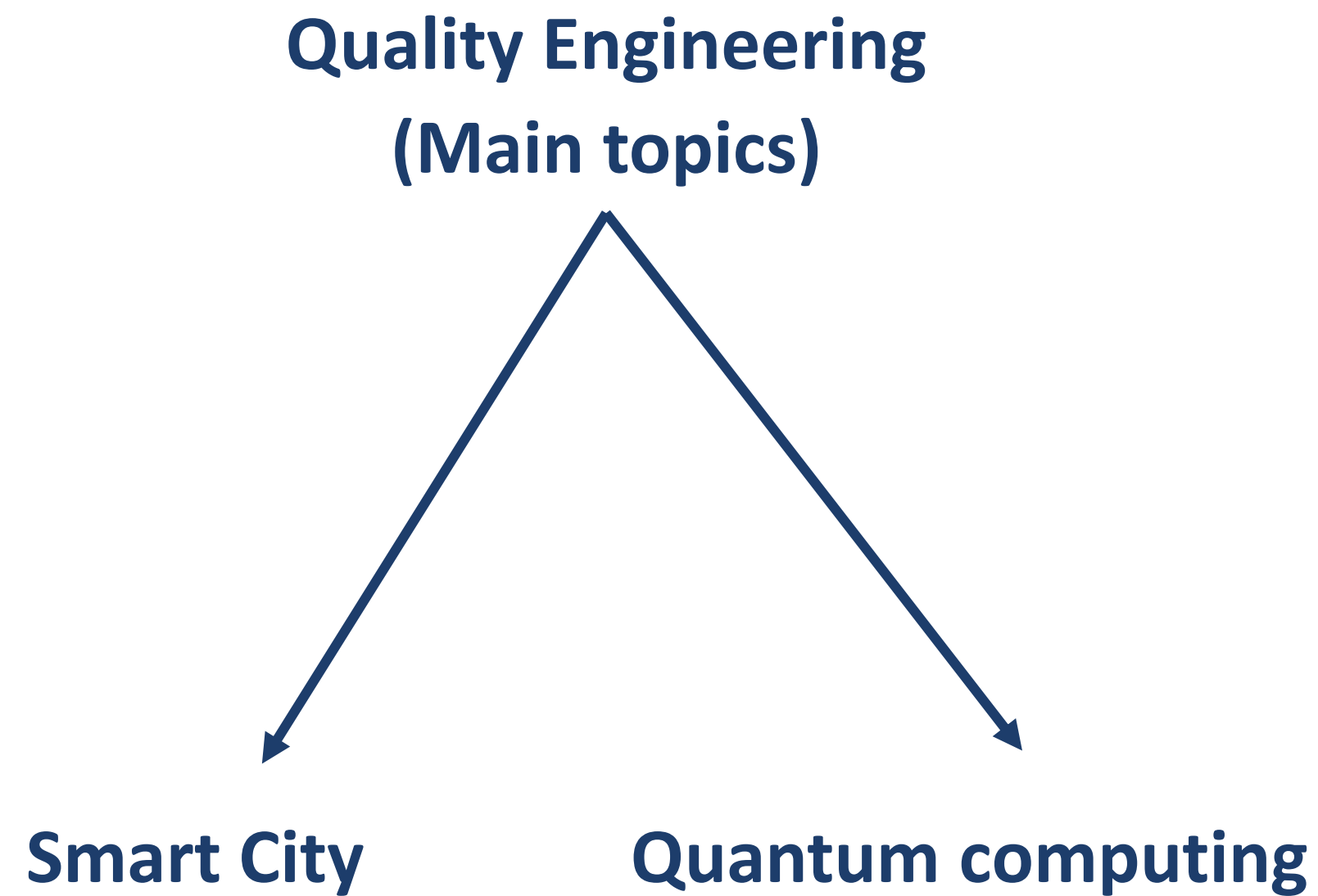
Funded by the European Union

Use case 4: Eclipse-Qrisp for PQC

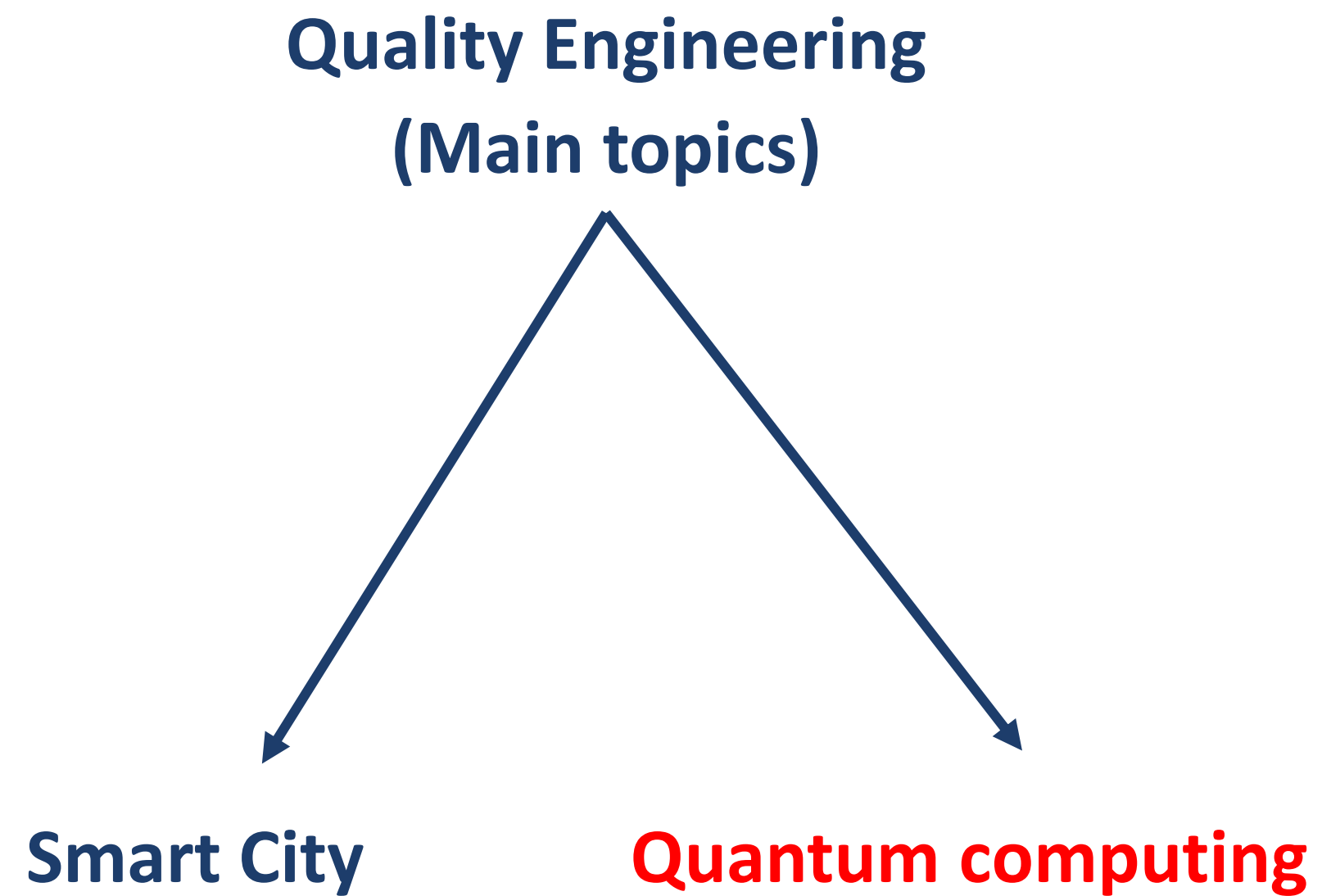
Contact at FHG:

- Dr. Rene Zander, rene.zander@fokus.fraunhofer.de
- Dr. Tobias Köppl, tobias.koeppl@fokus.fraunhofer.de

Quality Engineering at FHG



Quality Engineering at FHG



Use Case for open call

- **Use Case 4: Eclipse-Qrisp for PQC**
- **Task:** Design and build a framework using Qrisp to identify the best parameters to be used within PQC algorithms (in particular lattice based cryptosystems)

Post-Quantum Cryptography

- Cryptographic systems that are secure against both classical and quantum computers
- Key-encapsulation mechanisms (KEM)
 - Algorithms to establish a shared secret key over a public channel
 - Shared key used for symmetric-key cryptographic algorithms

PQC algorithms

- Security of public-key cryptographic methods relies on hardness to solve certain mathematical problems:
 - E.g., integer factorization problem (for RSA, broken)
- Proposed schemes:
 - Code-based encryption
 - Lattice-based encryption/signatures
 - Multivariate-quadratic-equation signatures
 - Hash-based signatures

Eclipse Qrisp



The next generation of quantum algorithm development

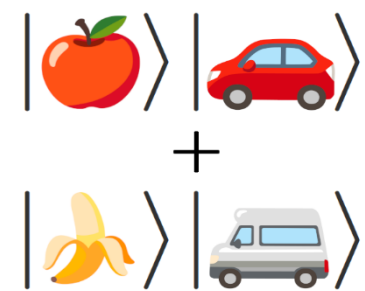
Qrisp is a high-level programming language for creating and compiling quantum algorithms. Its structured programming model enables scalable development and maintenance.

Eclipse Qrisp - Algorithms and algorithmic primitives modules

- Continuously growing libraries of **algorithmic primitives**: Quantum Fourier Transform (QFT), Quantum Phase Estimation (QPE), Quantum Amplitude Estimation, etc.
- **Algorithms**: Shor's algorithm, Quantum backtracking, QAOA etc.
- Utilize quantum algorithms in Qrisp for testing and validation of PQC methods against quantum (or hybrid) attacks

Qrisp - *what was that, again?*

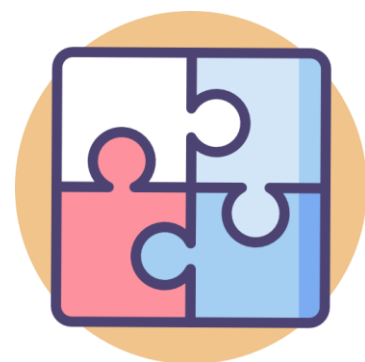
- High-level quantum programming language
- User friendly + built on python = **little to no learning overhead**
- Ever growing amount of **state-of-the-art features**:



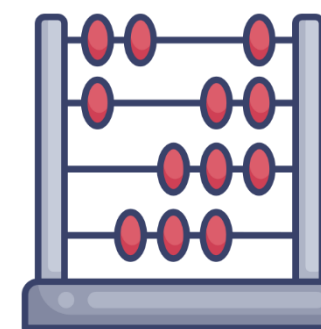
Typed QuantumVariables



Automatic Uncomputation



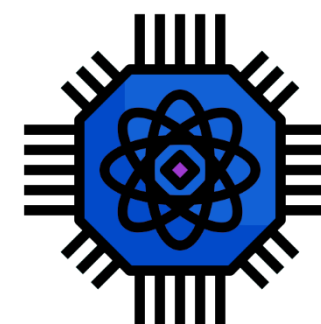
Modularity



Arithmetic



Compatibility



Simulator

Qrisp - Quantum Variables

Data types in Qrisp:

- QuantumFloat
- QuantumChar
- QuantumBool
- QuantumString

QRISP - Example



```
from qiskit import (QuantumCircuit, QuantumRegister,
                    ClassicalRegister, Aer, execute)
from qiskit.circuit.library import RGQFTMultiplier
n = 6
a = QuantumRegister(n)
b = QuantumRegister(n)
res = QuantumRegister(2*n)
cl_res = ClassicalRegister(2*n)
qc = QuantumCircuit(a, b, res, cl_res)
for i in range(len(a)):
    if 3 & 1<<i: qc.x(a[i])
for i in range(len(b)):
    if 4 & 1<<i: qc.x(b[i])
qc.append(RGQFTMultiplier(n, 2*n),
          list(a) + list(b) + list(res))
qc.measure(res, cl_res)
backend = Aer.get_backend('qasm_simulator')
counts_dic = execute(qc, backend).result().get_counts()
print({int(k, 2) : v for k, v in counts_dic.items()})
#Yields: {12: 1024}
```

```
from qrisp import QuantumFloat
n = 6
a = QuantumFloat(n)
b = QuantumFloat(n)
a[:] = 3
b[:] = 4
res = a*b
print(res)
#Yields: {12: 1.0}
```


QRISP - Example



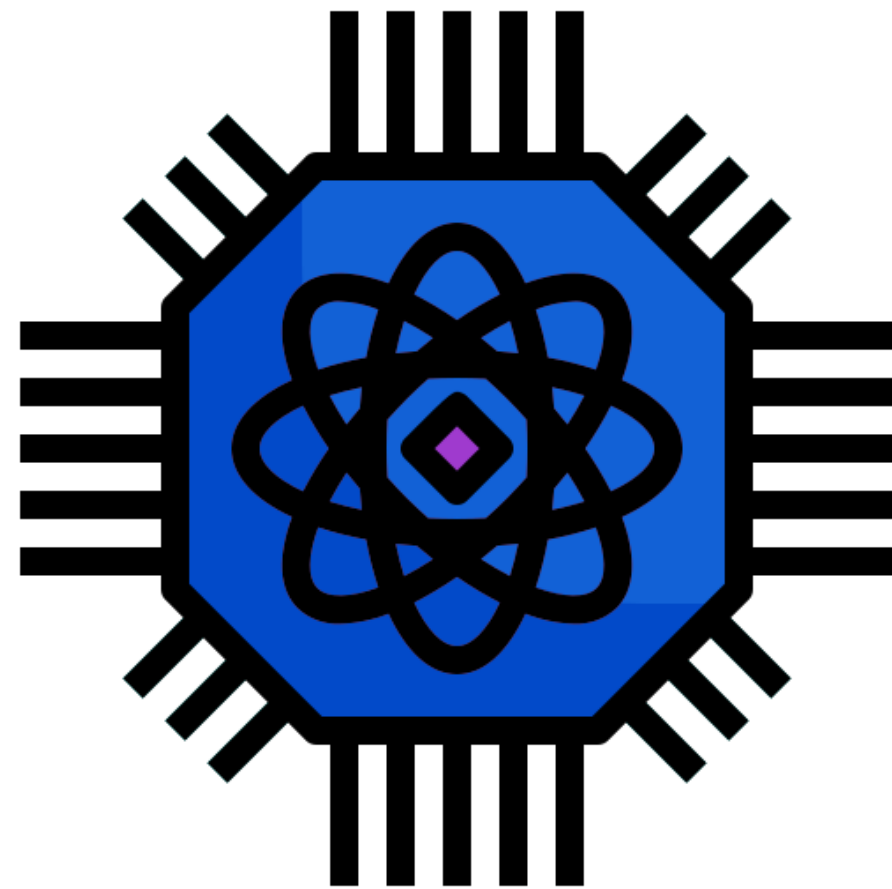
```
from qiskit import (QuantumCircuit, QuantumRegister,
                    ClassicalRegister, Aer, execute)
from qiskit.circuit.library import RGQFTMultiplier
n = 6
a = QuantumRegister(n)
b = QuantumRegister(n)
res = QuantumRegister(2*n)
cl_res = ClassicalRegister(2*n)
qc = QuantumCircuit(a, b, res, cl_res)
for i in range(len(a)):
    if 3 & 1<<i: qc.x(a[i])
for i in range(len(b)):
    if 4 & 1<<i: qc.x(b[i])
qc.append(RGQFTMultiplier(n, 2*n),
          list(a) + list(b) + list(res))
qc.measure(res, cl_res)
backend = Aer.get_backend('qasm_simulator')
counts_dic = execute(qc, backend).result().get_counts()
print({int(k, 2) : v for k, v in counts_dic.items()})
#Yields: {12: 1024}
```

```
from qrisp import QuantumFloat
n = 6
a = QuantumFloat(n)
b = QuantumFloat(n)
a[:] = 3
b[:] = 4
res = a*b
print(res)
#Yields: {12: 1.0}
```

On the 127 qubit ibm_osaka backend:

{17: 0.0216, 25: 0.0205, 57: 0.0201, 1: 0.0196,
49: 0.0196, 33: 0.0192, 3: 0.019, 53: 0.019, 51:
0.0185, 56: 0.0183, 8: 0.0181, 16: 0.0176, 19:
0.0176, 40: 0.0176, 9: 0.0175, 24: 0.0174, 43:
0.017, 44: 0.017,...}

Qrisp - Qrisp Simulator



- Qrisp comes with high-performance simulator
- utilizes sparse matrices to store and process quantum states
- (some) quantum circuits with 100+ qubits can be simulated

qrisp



qrisp.eu

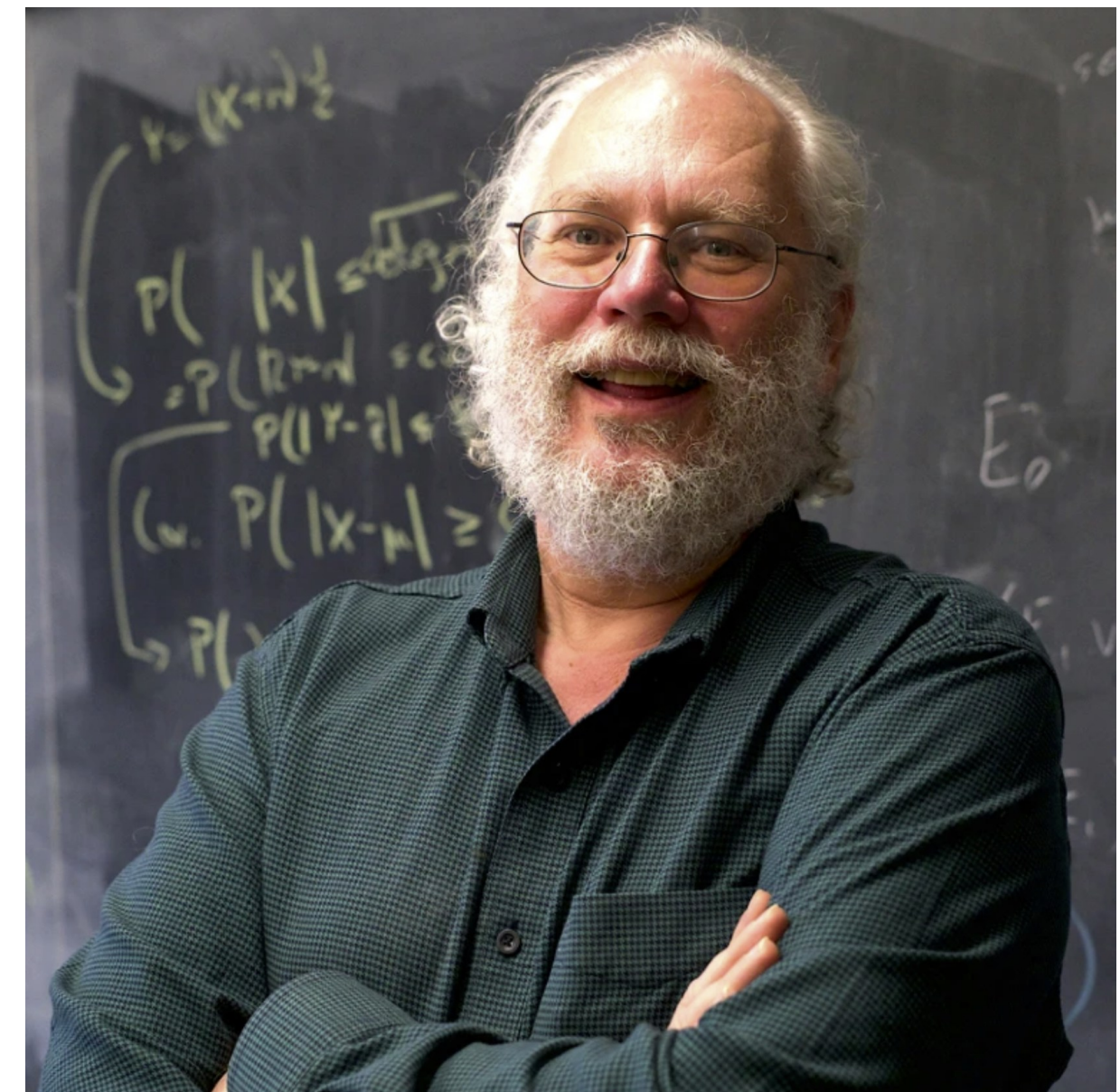


<https://github.com/eclipse-qrisp/Qrisp>

Shor's algorithm in Qrisp

Shor's factoring algorithm implementation

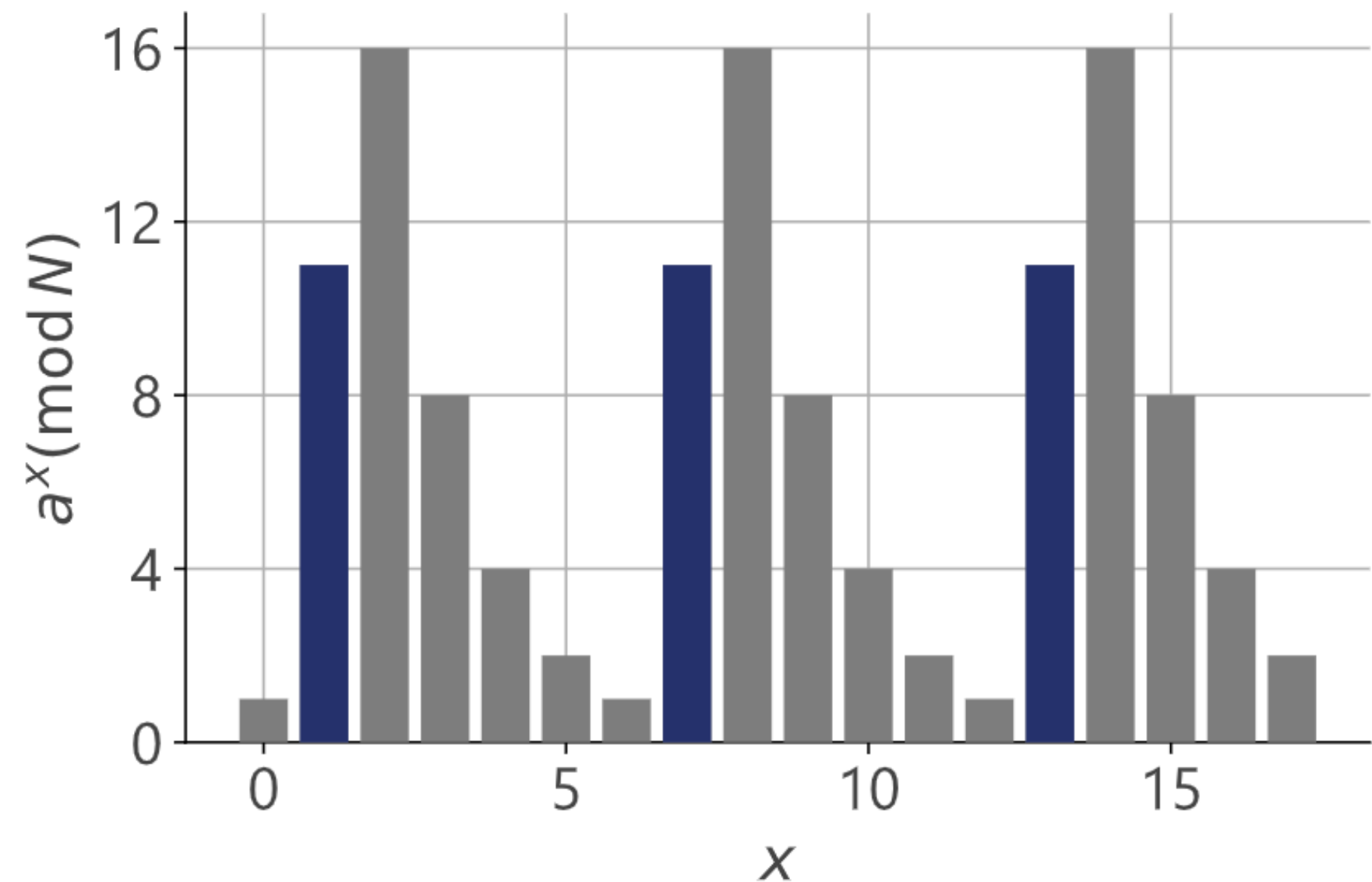
- **Factorizing numbers** exponentially faster than known classical algorithms (**exponential speedup**)
- Potential to factor large numbers within **reasonable timeframes**
- **Few implementations** across the framework landscape mostly utilizing a trick to factor $N=15$



Peter Shor; Credit: BBVA FOUNDATION

Shor's factoring algorithm - What?

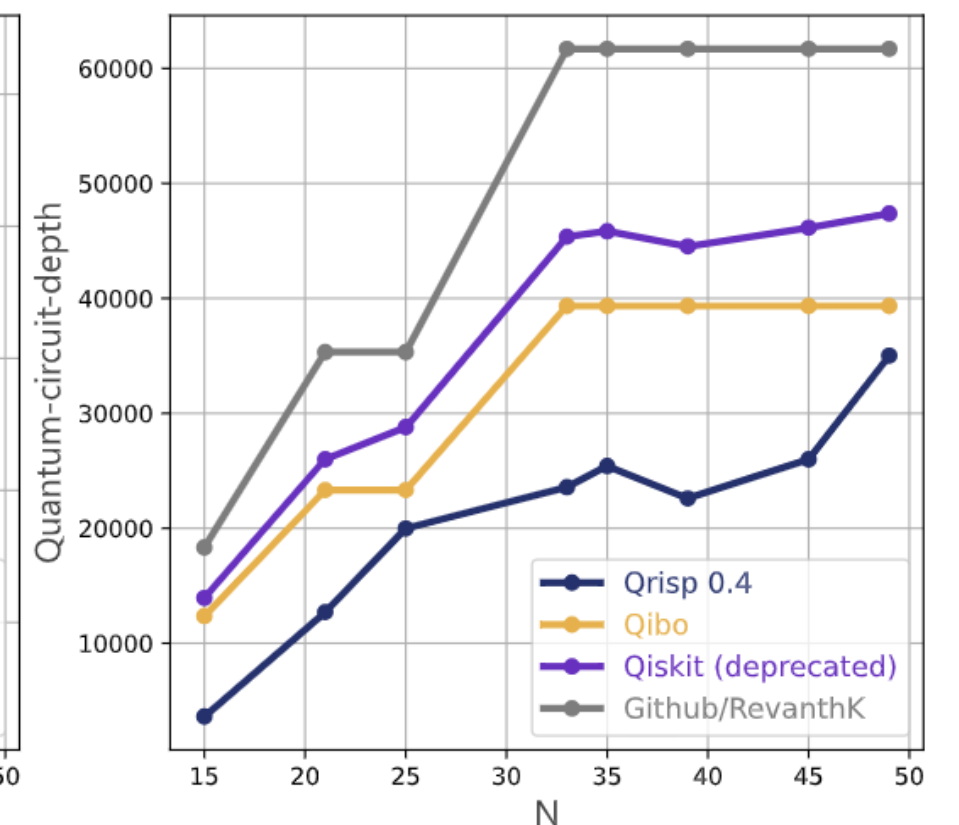
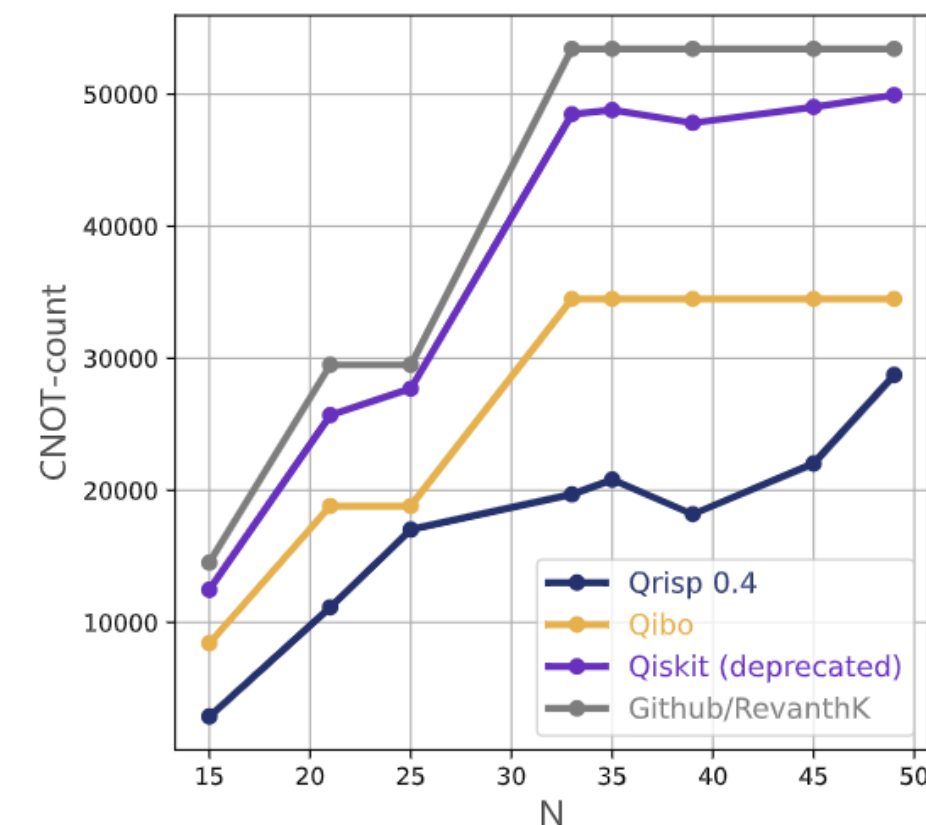
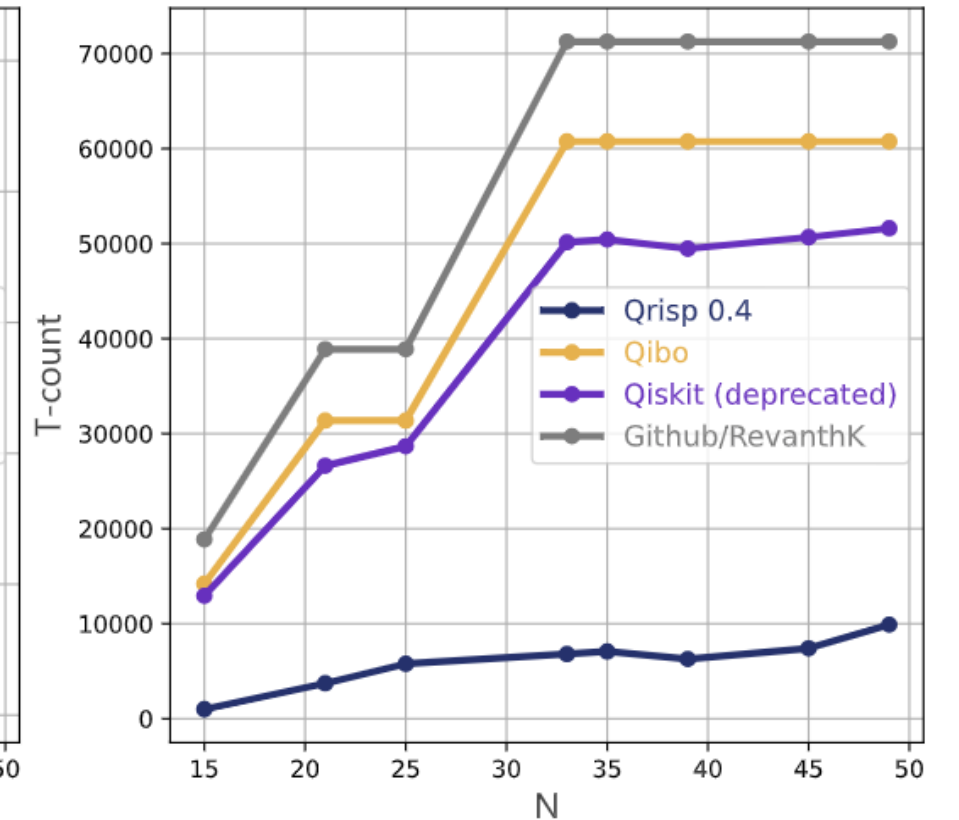
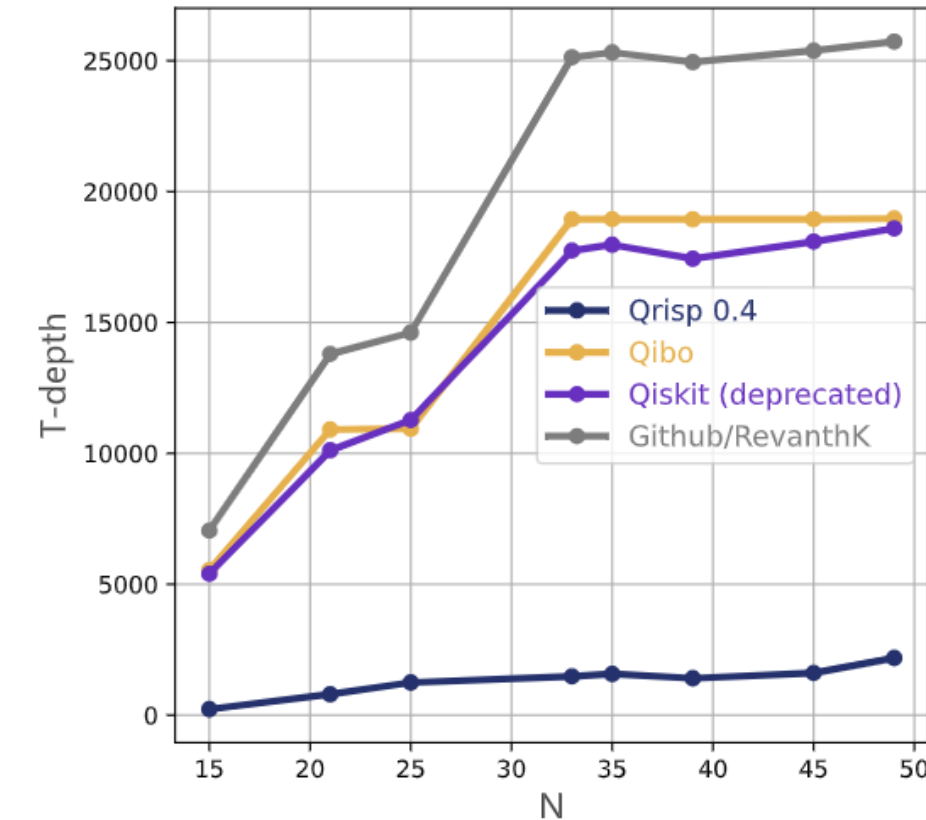
1. Select a random integer a smaller than N (number we want to factorize)
2. Calculate **greatest common divisor** classically to check for accidental success (less and less likely with increasing N)
3. Why quantum?
→ Determine the **period of a modulo N** (not easily achievable classically)
4. Find non-trivial factor of N or select new random integer



Shor's factoring algorithm - How?

- Sounds simple? → VERY DIFFICULT TO IMPLEMENT!
- Only 3 other implementations found online – mostly factoring $N=15$ (smallest instance)
- With Qrisp one can factor larger numbers as simply as calling

```
from qrisp.shor import shors_alg  
  
shors_alg(65)
```



Implementation and testing of PQC algorithms (ML-KEM)

- Module-Lattice-based Key-Encapsulation Mechanism Standard (ML-KEM)
- Description of the algorithm:
FIPS 203 (Federal Information Processing Standards Publication, NIST)

NIST Report (FIPS 203)



1 FIPS 203 (Draft)

2 Federal Information Processing Standards Publication

3

4 Module-Lattice-based 5 Key-Encapsulation 6 Mechanism Standard

7 Category: Computer Security

Subcategory: Cryptography

8 Information Technology Laboratory
9 National Institute of Standards and Technology
10 Gaithersburg, MD 20899-8900

11 This publication is available free of charge from:
12 <https://doi.org/10.6028/NIST.FIPS.203.ipd>

13 Published August 24, 2023



14

Funded by the European Union

Challenges for a possible project

Solve the underlying
mathematical problem
using quantum
algorithms

Resistance of
implementations
against side-
channel attacks

Computational
efficiency, key sizes

References

National Institute of Standards and Technology, FIPS 203, Module-Lattice-Based Key-Encapsulation Mechanism Standard, <https://csrc.nist.gov/pubs/fips/203/ipd>, 2023.

D. Bernstein and T. Lange, Post-quantum cryptography, *Nature*, vol. 549, pp. 188--194, 2017.

D. Dachman-Soled, L. Ducas, H. Gong, M. Rossi, LWE with Side Information: Attacks and Concrete Security Estimation, *Advances in Cryptology--CRYPTO 2020*, Springer International Publishing, 2020.

F. Göpfert, C. van Vredendaal and T. Wunderer, A hybrid lattice basis reduction and quantum search attack on LWE, In *Post-Quantum Cryptography: 8th International Workshop, PQCrypto 2017, Proceedings 8* (pp. 184--202). Springer International Publishing, 2017.

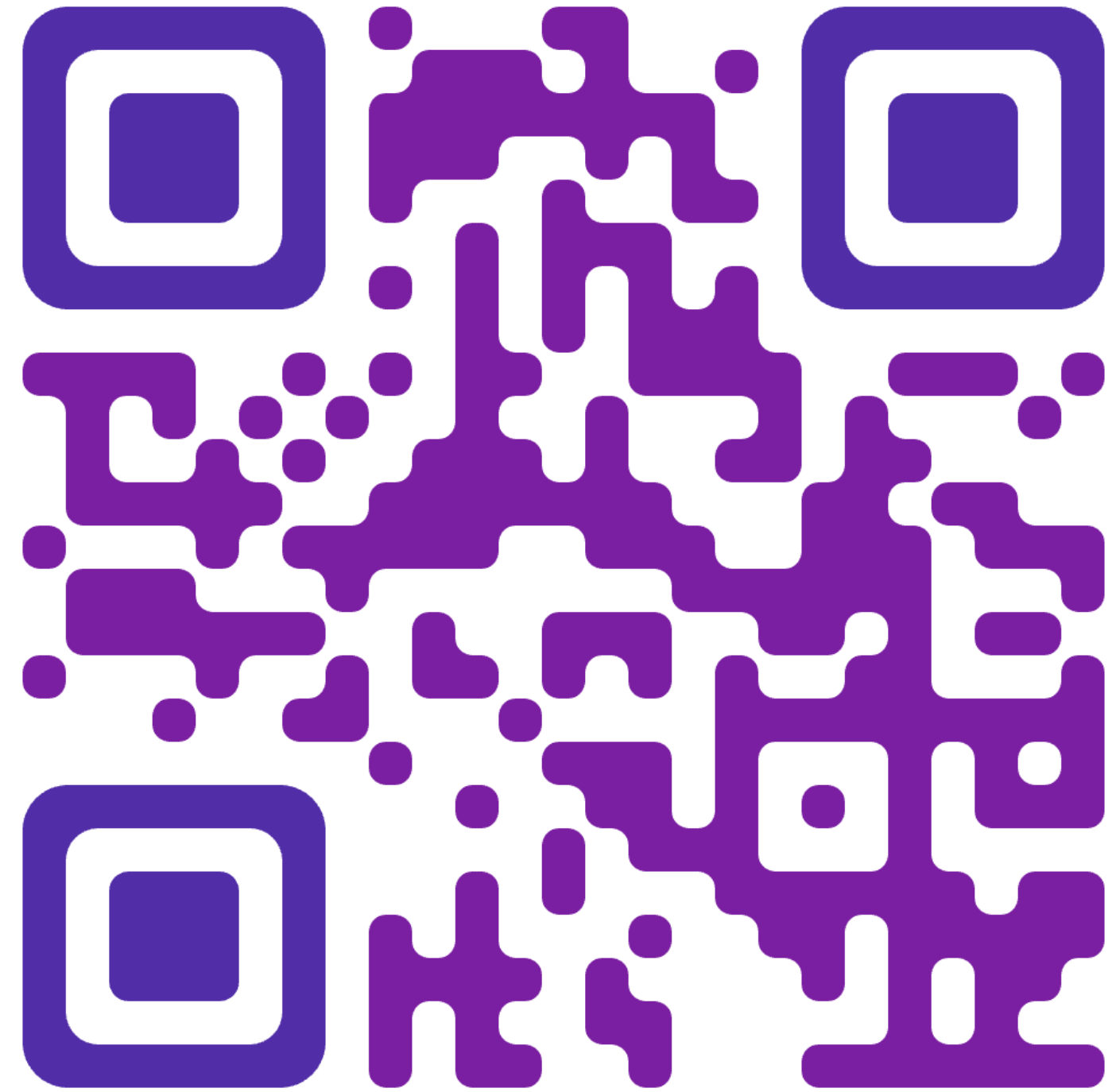
T. Köppl, R. Zander and N. Tcholchev. Resilience of lattice-based Cryptosystems to Quantum Attacks. In *2024 IEEE Symposium on Computers and Communications (ISCC)* (pp. 1-6). IEEE, 2024.

T. Köppl, R. Zander, L. Henkel and N. Tcholchev. A parameter study for LLL and BKZ with application to shortest vector problems. *arXiv preprint arXiv:2502.05160*, 2025.

Thank you for your attention!

<https://www.linkedin.com/company/pq-react/>

<https://pqreact.eu>





How to apply?

PQ-REACT Helpdesk

Helpdesk email:

applications@pqreact.eu

Website:

<https://pqreact.eu/open-call-2/>



Q&A



pq·react



@PQREACT



@PQ-REACT



@PQ-REACT-EU-PROJECT



Funded by
the European Union

Thank you!



pq·react



@PQREACT



@PQ-REACT



@PQ-REACT-EU-PROJECT



Funded by
the European Union



pq·react

Open call 2 - IMPLEMENT

2nd Pitching session

29/04/2025



Funded by
the European Union

AGENDA

- 1. 12:00h - 12:15h Round table**

Thank you!



pq·react



@PQREACT



@PQ-REACT



@PQ-REACT-EU-PROJECT



Funded by
the European Union