ECCentric

An Empirical Analysis of Quantum Error Correction Codes

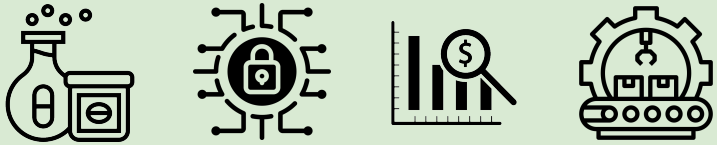
Aleksandra Świerkowska, Jannik Pflieger,
Emmanouil Giortamis, Pramod Bhatotia

Technical University of Munich

ACM SIGMETRICS 2026, Ann Arbor, USA

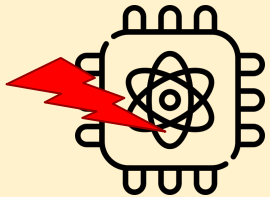
State of quantum computing

Quantum applications

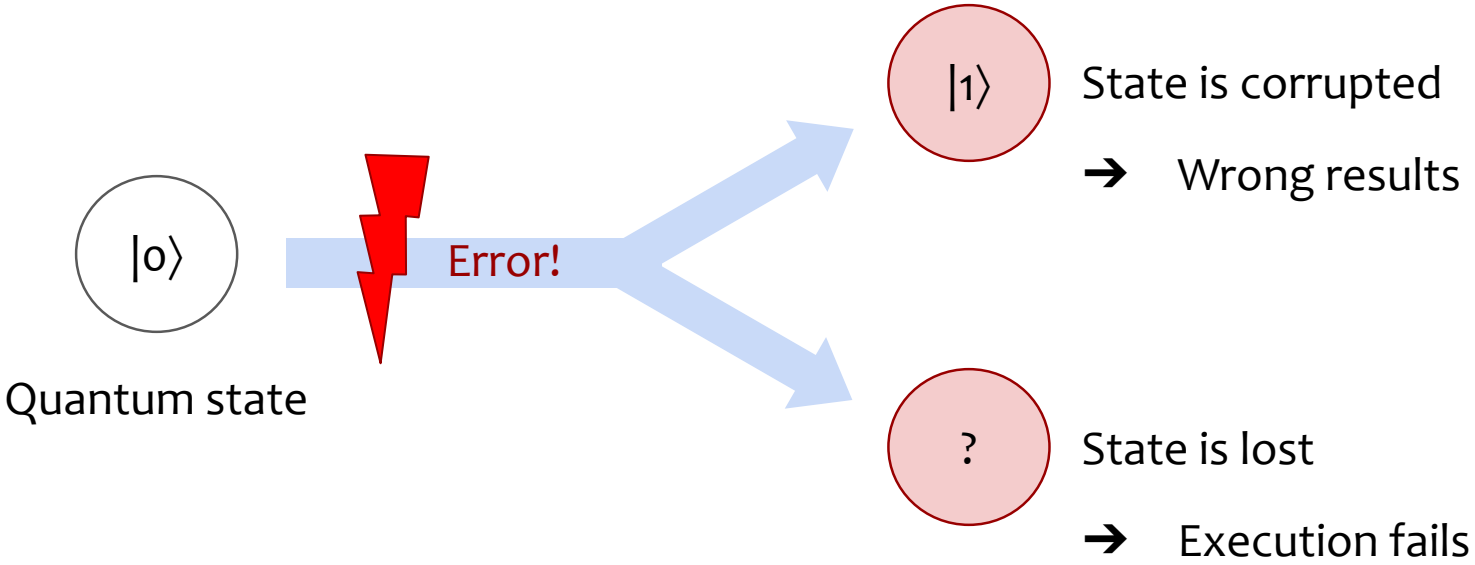


Quantum Processing Unit (QPU)

Errors!



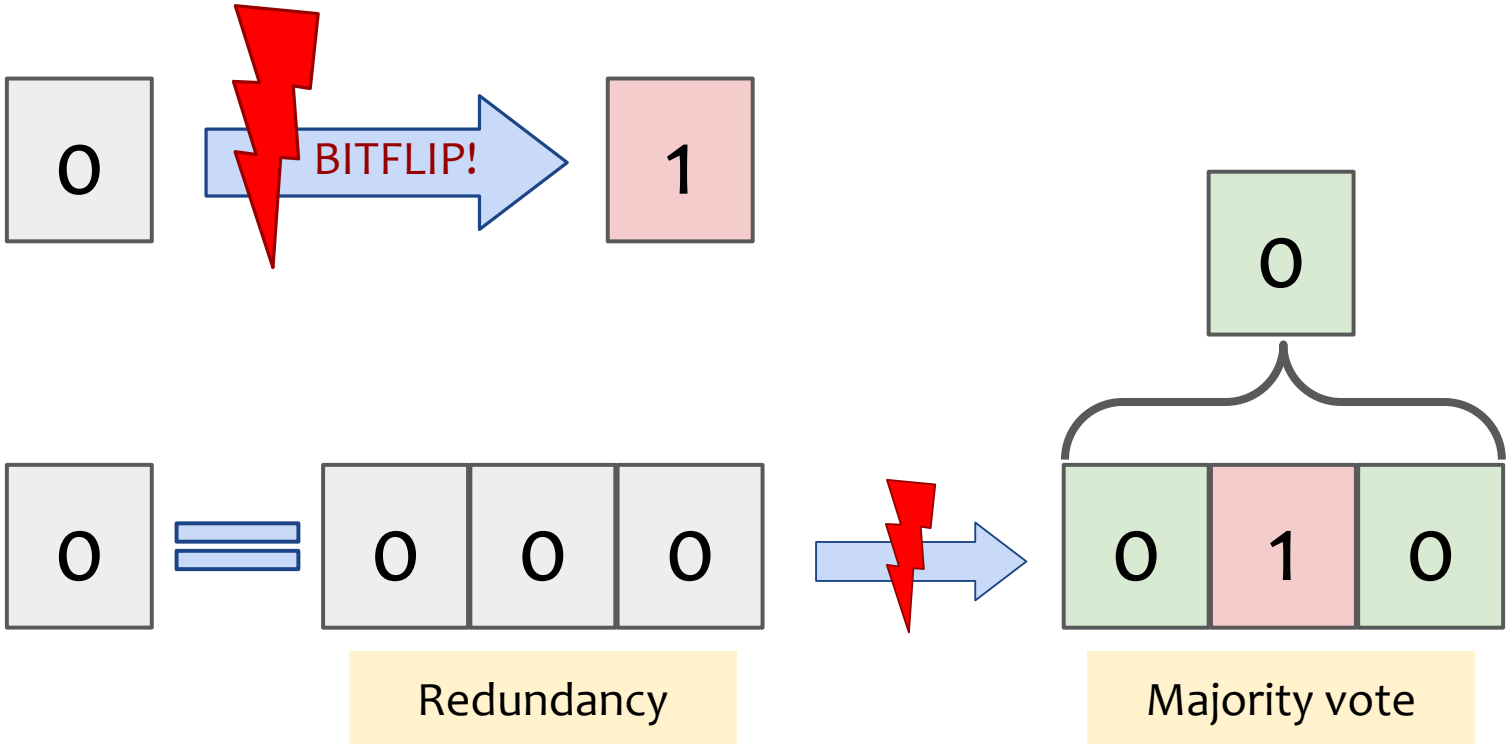
What is an error?



How can we protect quantum computations from errors?

Quantum error correction (QEC)

How does *“classical”* error correction work?

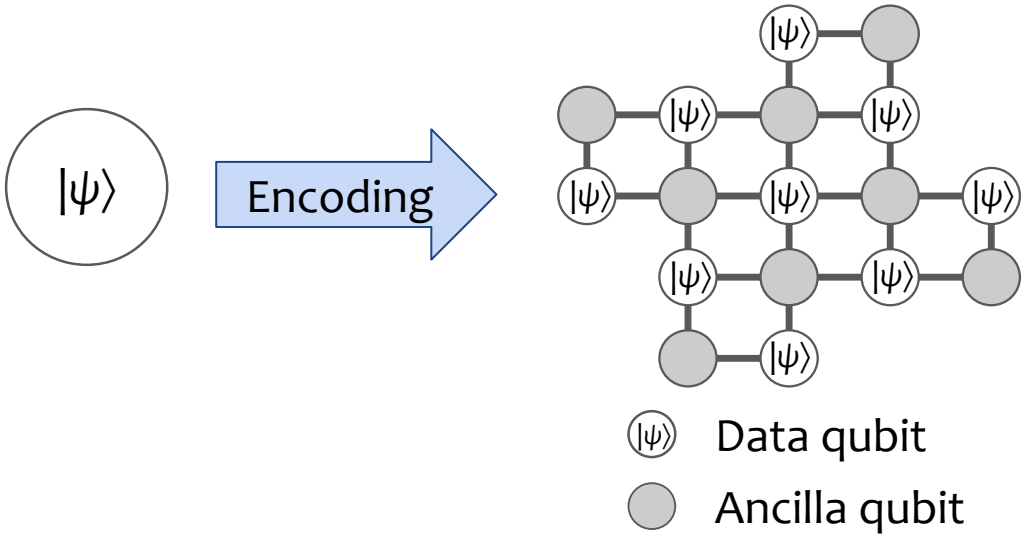


Is quantum error correction just as simple?

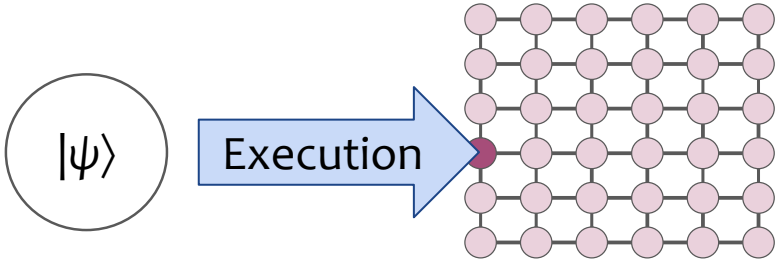
Unfortunately, not!

Physics forbids copying quantum states, and reading destroys them

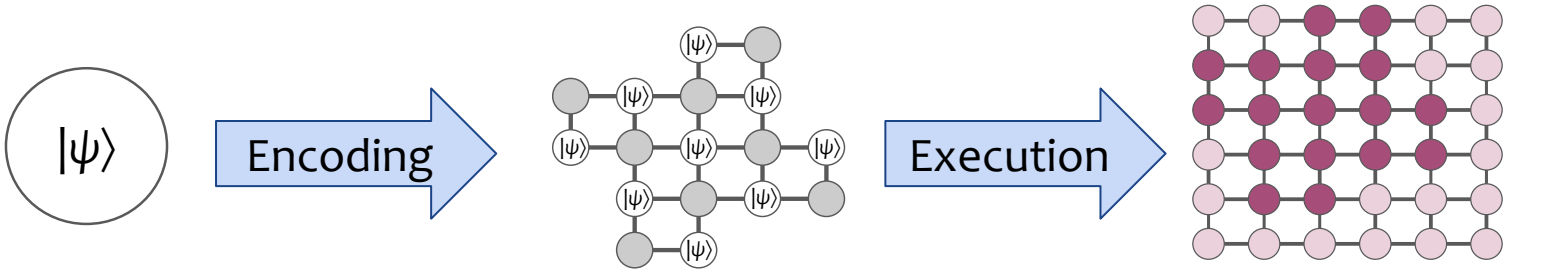
How can we circumvent the issue of reading state?



How does quantum error correction work?

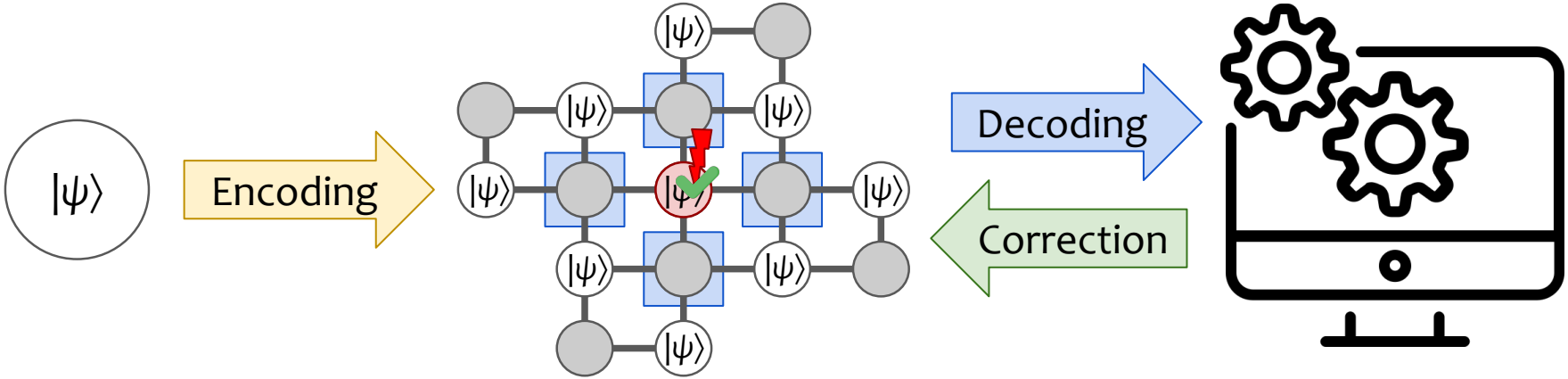


Application qubit **Physical qubits**



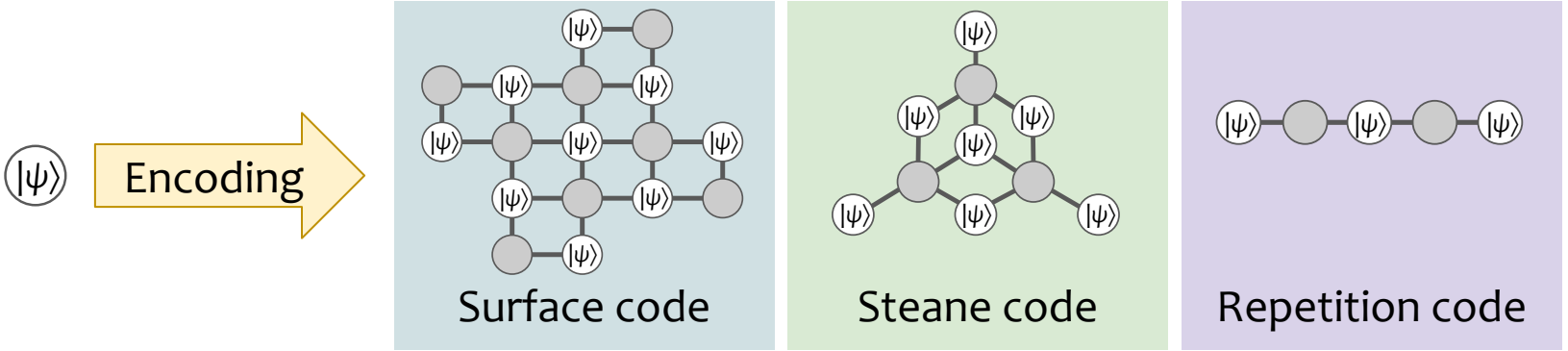
Application qubit **Logical qubit** **Physical qubits**

How can we detect errors?



Key part of the execution is the chosen QEC code

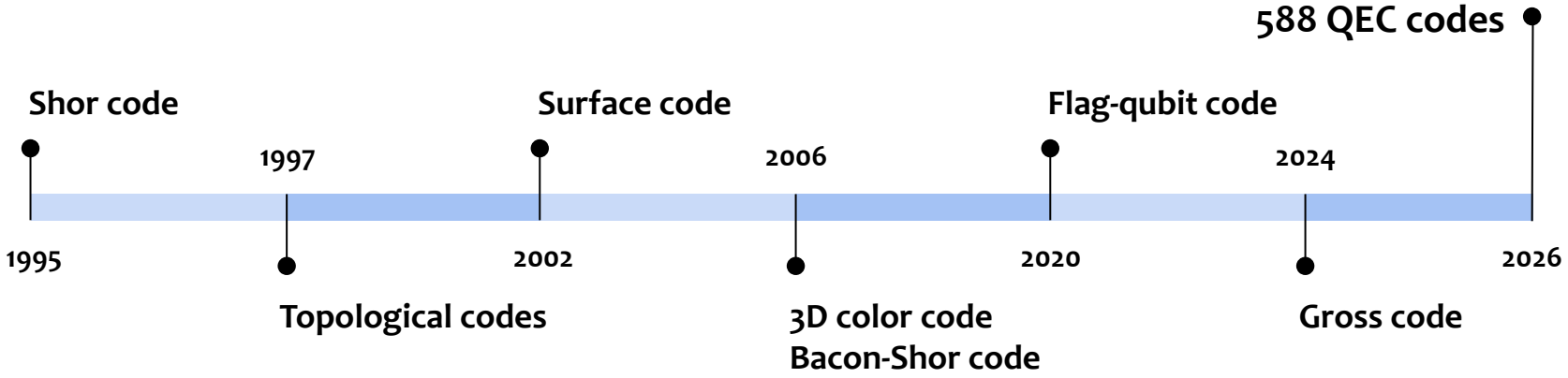
Quantum error correction codes



QEC codes can differ in:

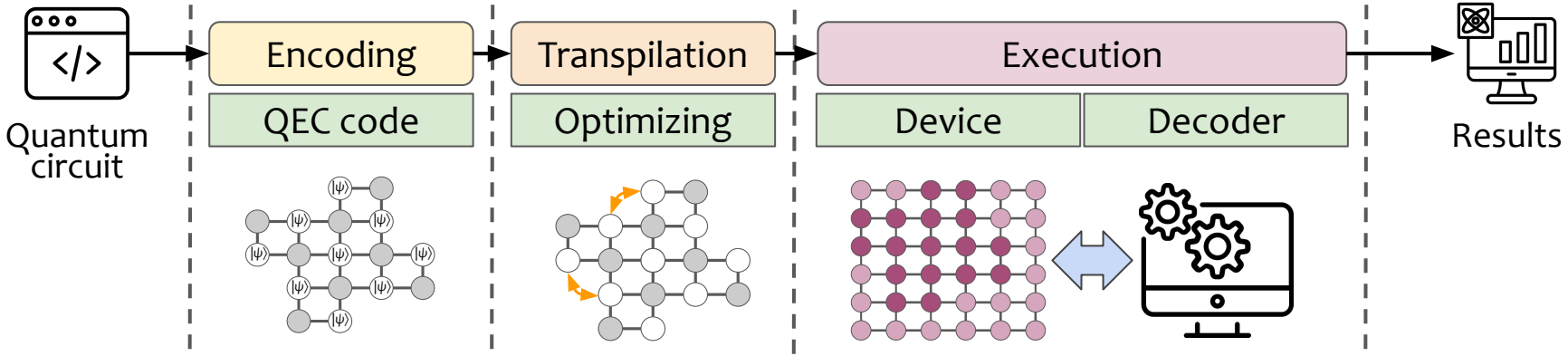
- Resource overhead
- Hardware requirements
- Noise threshold

Zoo of QEC codes



QEC codes must maintain compatibility throughout execution stages

Execution pipeline



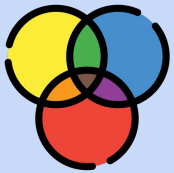
Motivation

There is no work with general, realistic evaluations of QEC codes

We provide a comprehensive empirical analysis of QEC codes

Challenges

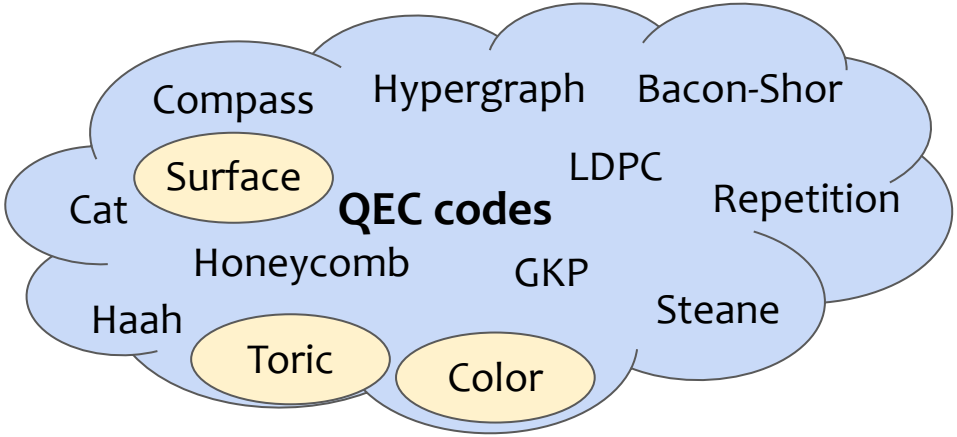
#1 Missing classification of codes



#2 Intractable exploration space

#3 Fragmentation of the QEC toolchain

Challenge #1: Missing classification of codes



Problem: There is no clear classification of QEC codes

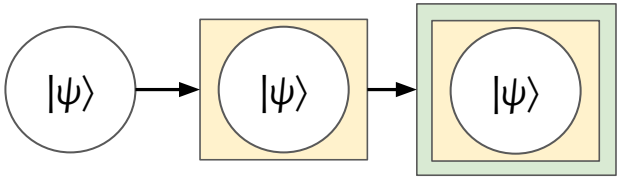
Key idea: Introducing QEC codes taxonomy

Insight #1: Shared characteristics

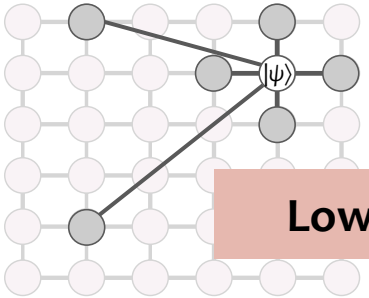
Mathematical properties of QEC codes

- Resource overhead
- Hardware requirements
- Noise threshold

Concatenation



+ good thresholds
– very large resource overhead

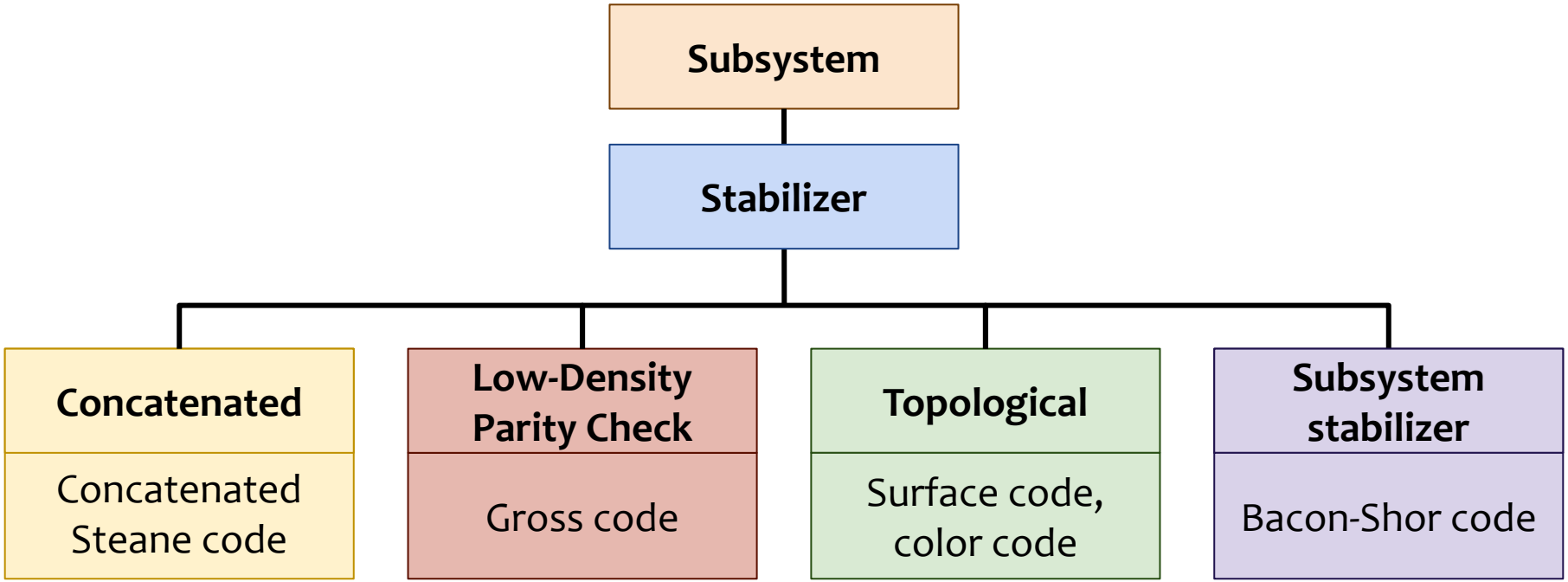


Low density

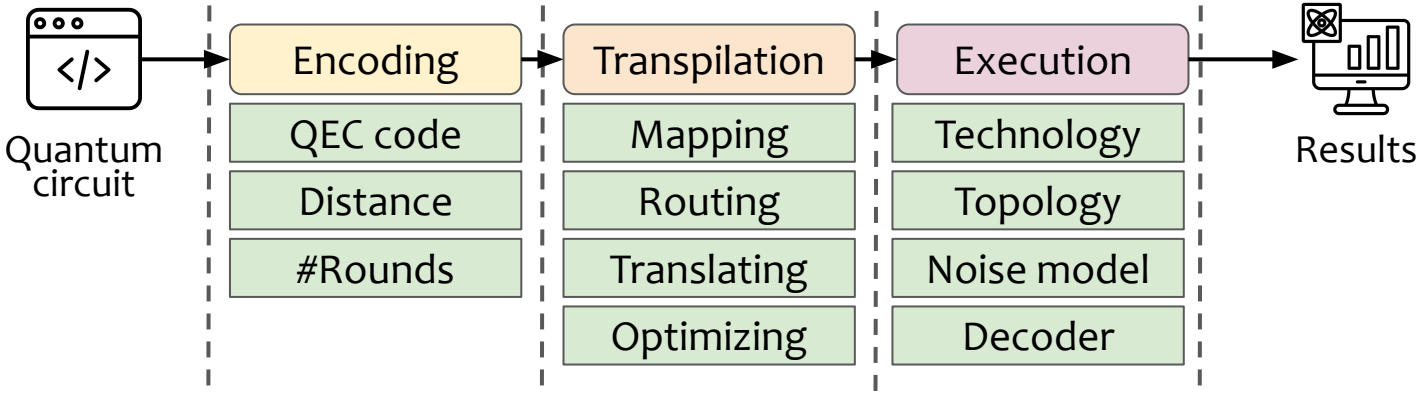
+ good thresholds
+ low overhead
– complex hardware requirements

Different groups of mathematical properties result in different tradeoffs

Key idea #1: Taxonomy



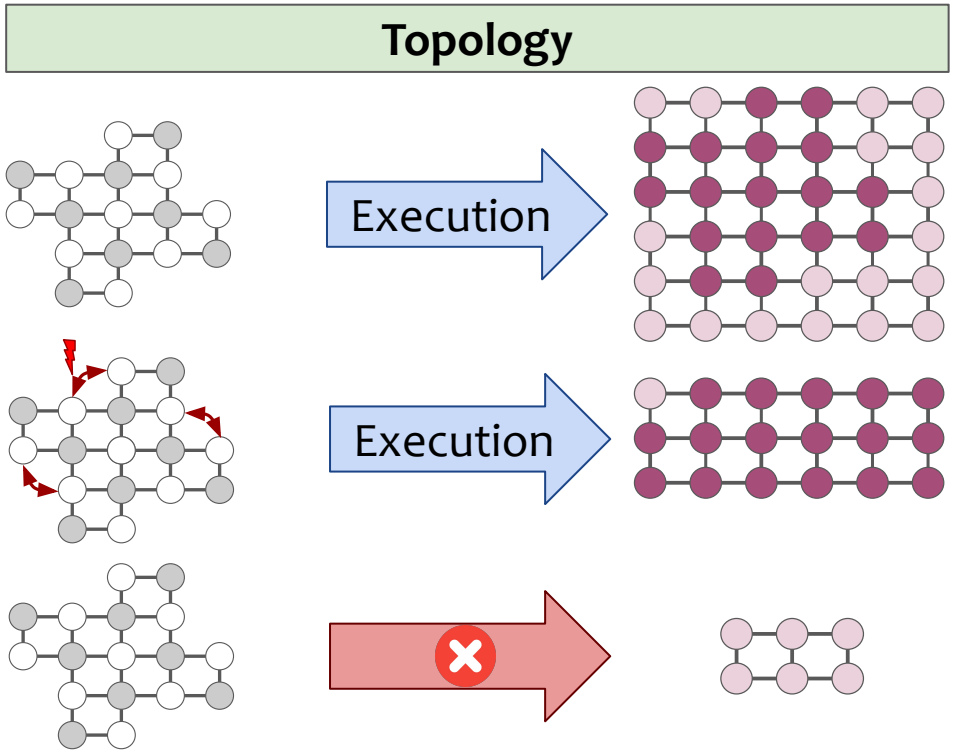
Challenge #2: Intractable exploration space



Problem: Each step of compilation and execution brings more dimensions

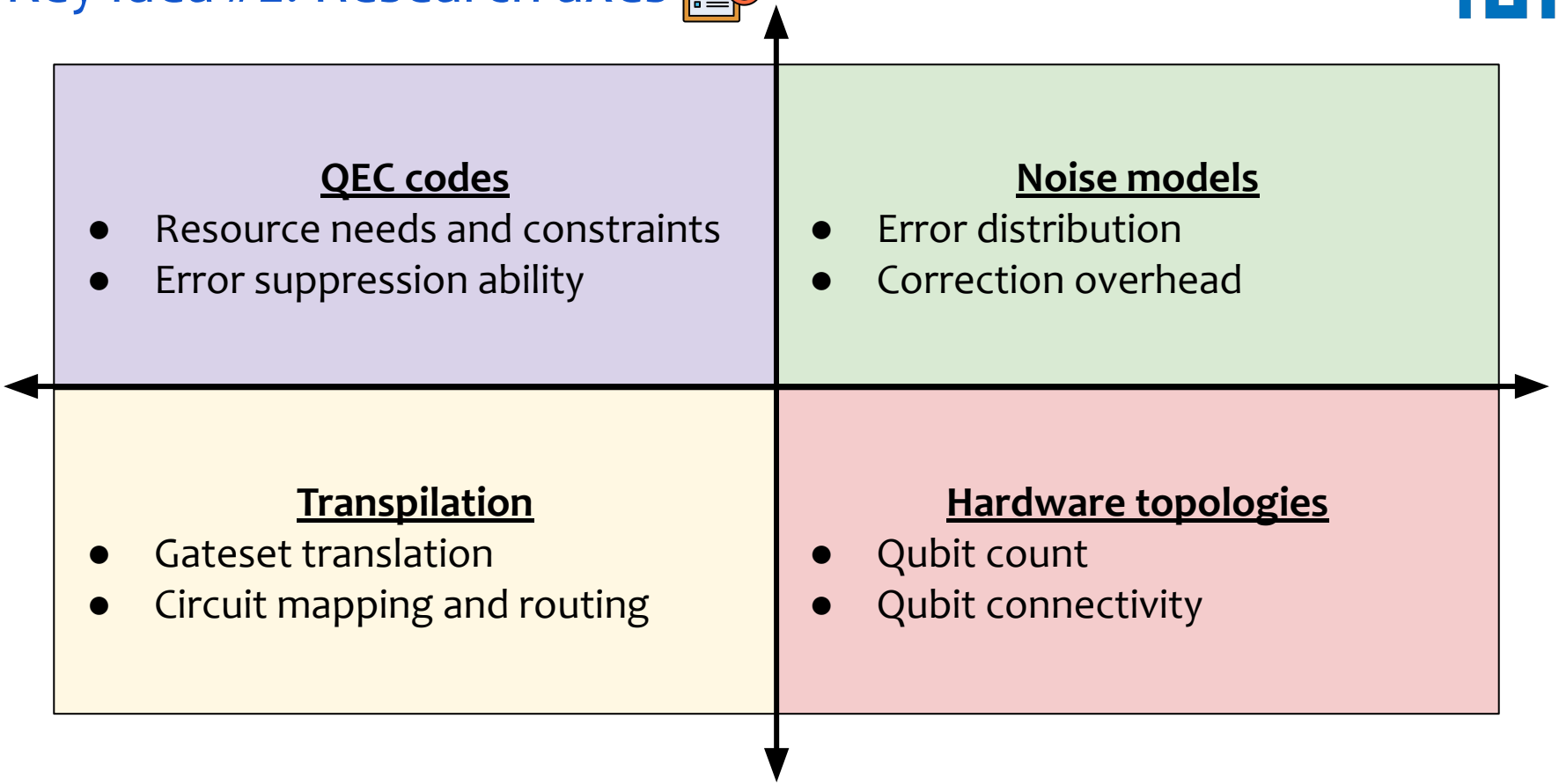
Key idea: Limiting the space by defining most critical research axes

Insight #2: Different limitations have different impact



Some parameters are significantly more important than others

Key idea #2: Research axes



Challenge #3: Fragmentation of the QEC toolchain

Qiskit

- Hardware-aware transpilation
- Detailed error modeling
- Slow simulation for large experiments

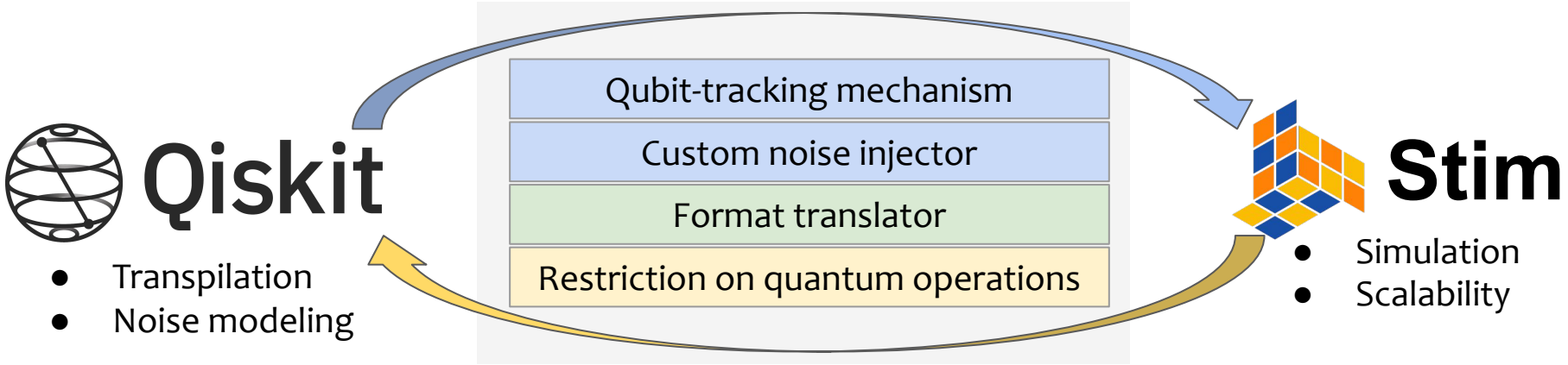
Stim

- Extremely fast simulation
- Scales to large QEC circuits
- No transpilation or device constraints

Problem: No single tool allowed us to perform realistic QEC analysis

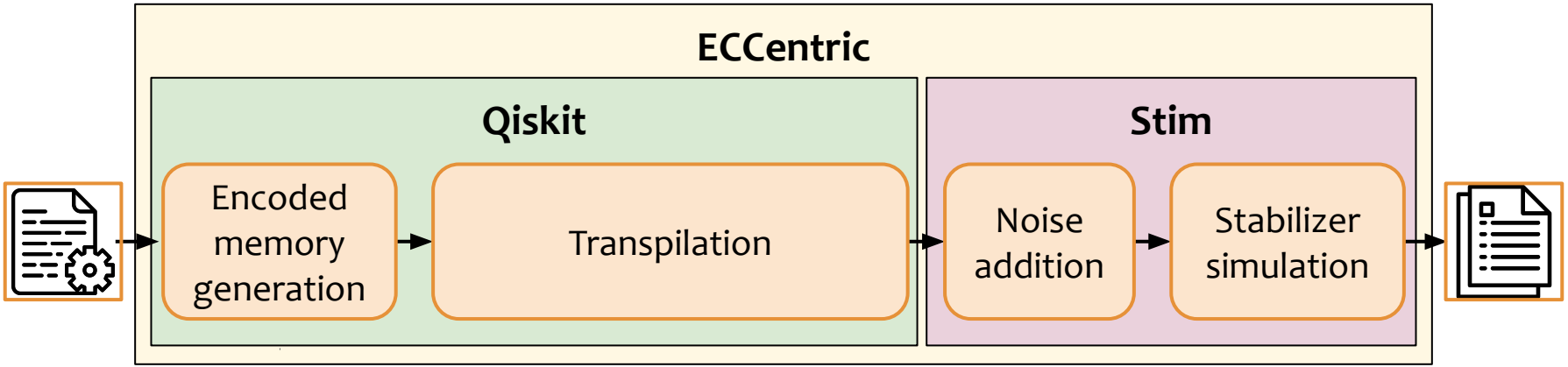
Key idea: Framework uniting detailed noise modeling, transpilation overhead with ultra-fast stabilizer simulation

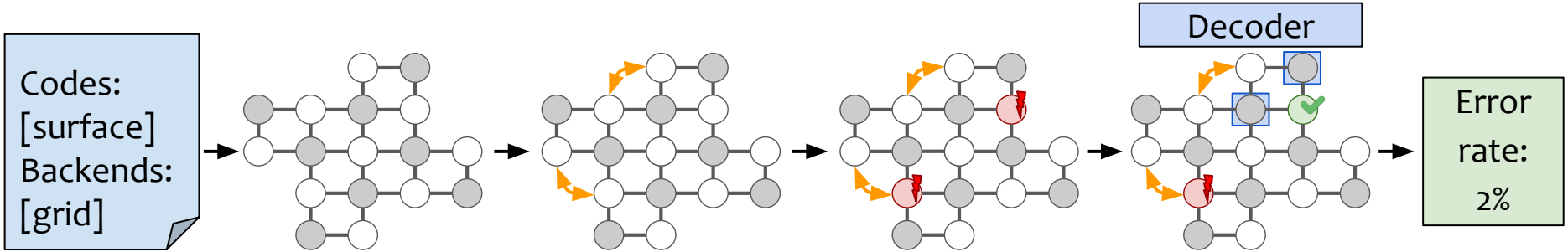
Insight #3: Best of both worlds



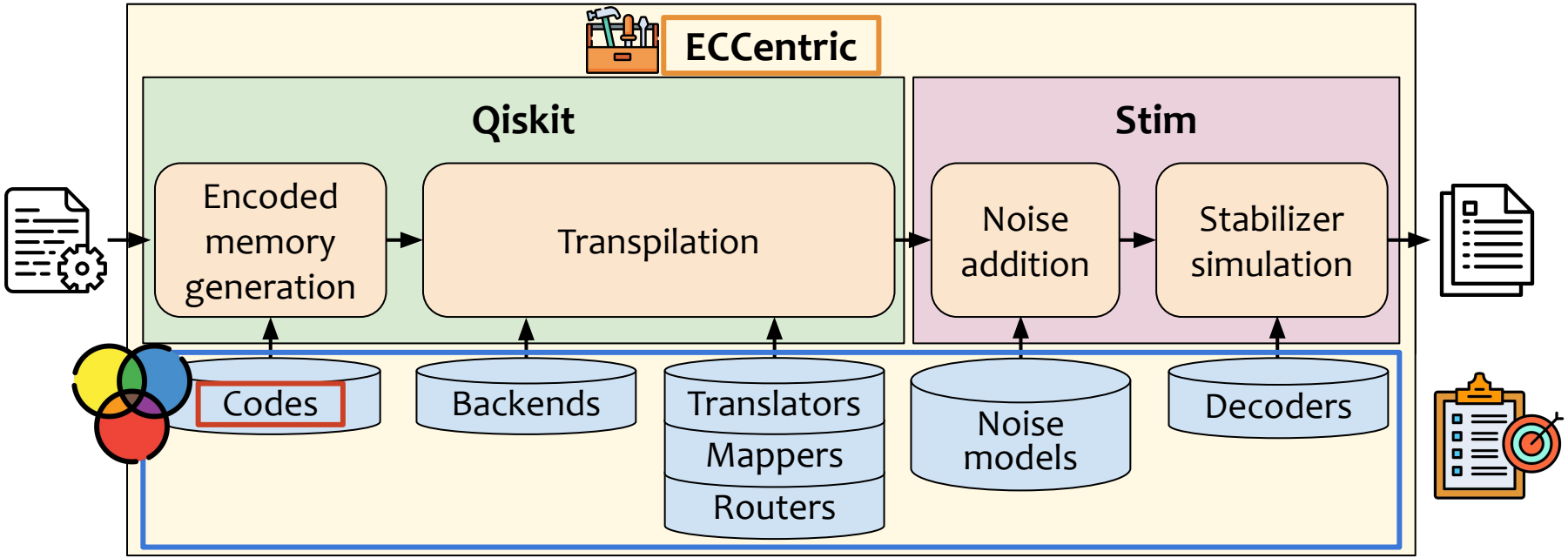
Our defined modules enable seamless integration of Qiskit and Stim

Key idea #3: ECCentric framework

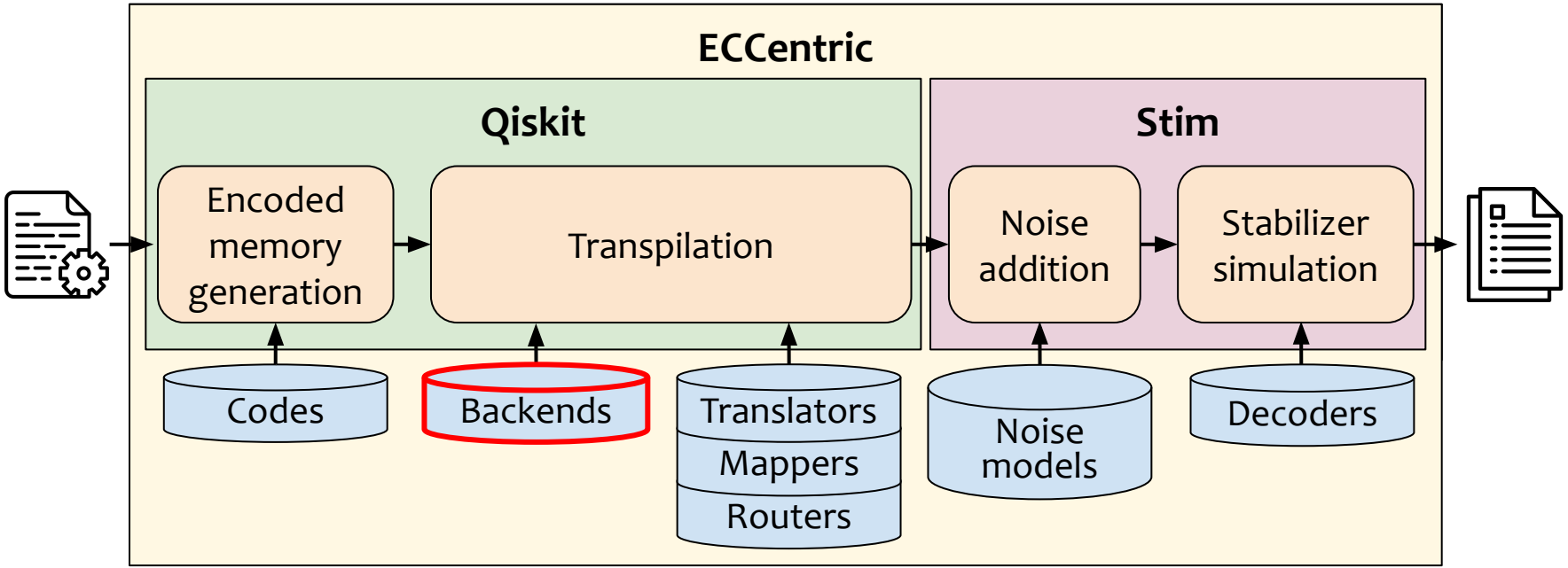




ECCentric overview



ECCentric overview



Today, we will mainly focus on the impact of backend

In this talk I will answer 3 questions

Q1: Does variance in qubit quality critically impact logical error rates?

Q2: Which quantum technology is most suitable for effective QEC?

Q3: Is applying QEC always beneficial?

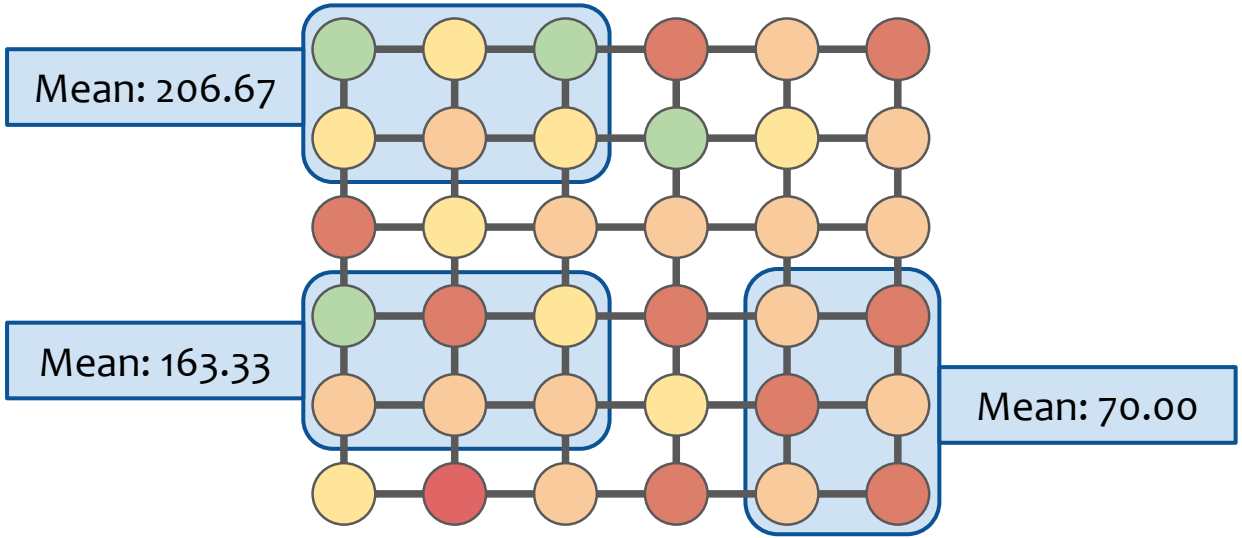
Please refer to the paper for the full evaluation:

Q1: Variance context

Decoherence time:

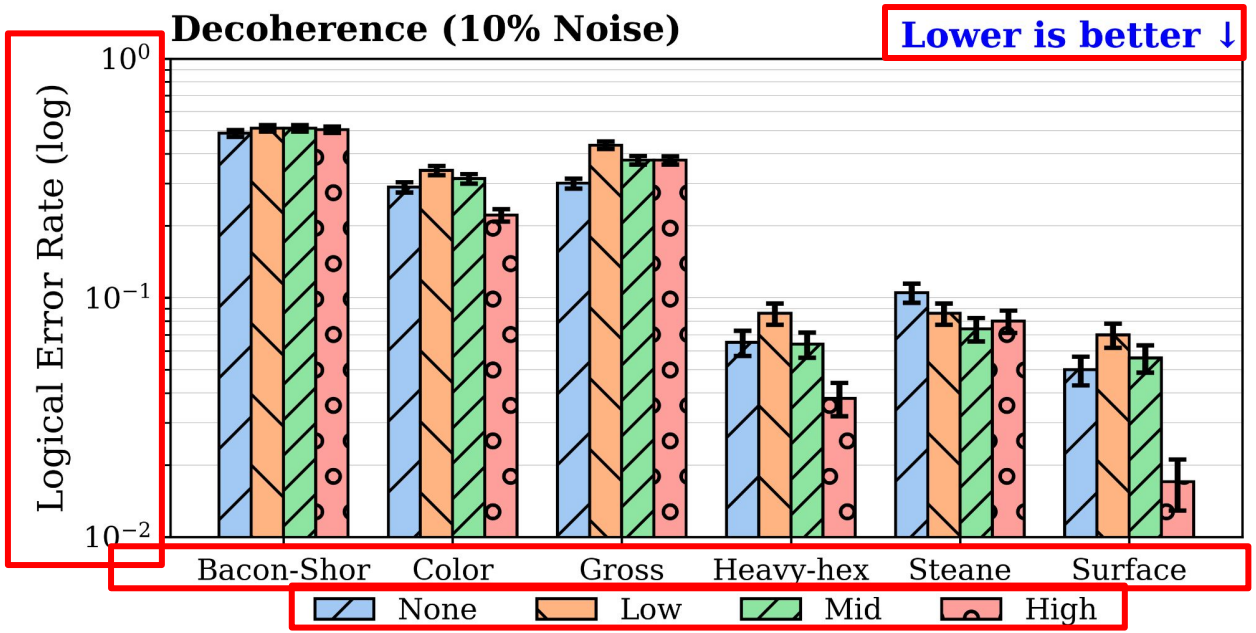
- 300 µs
- 180 µs
- 100 µs
- 40 µs

Mean decoherence time of the system: 123.33



Heterogeneity in qubit quality may strongly influence the error rate

Q1: Variance results



Changing qubit quality variance alters error rate by up to **4×** across codes, but without a consistent monotonic trend

None: 0 µs, low: 60 µs, mid: 120 µs, high: 180 µs

Q2: Technology context

	Superconducting	Trapped-ion	Neutral atoms
Topology	Heavy-hex, grid	Race track, grid	2D array, arbitrary 3D geometry
Shuttling	No	Yes	Yes
Coherence time	10 - 100 µs	4 - 3600 s	1 - 60s
Operation speed	12 - 25 ns	1 - 10 µs	0.4 - 2 µs

Q2: Technology context

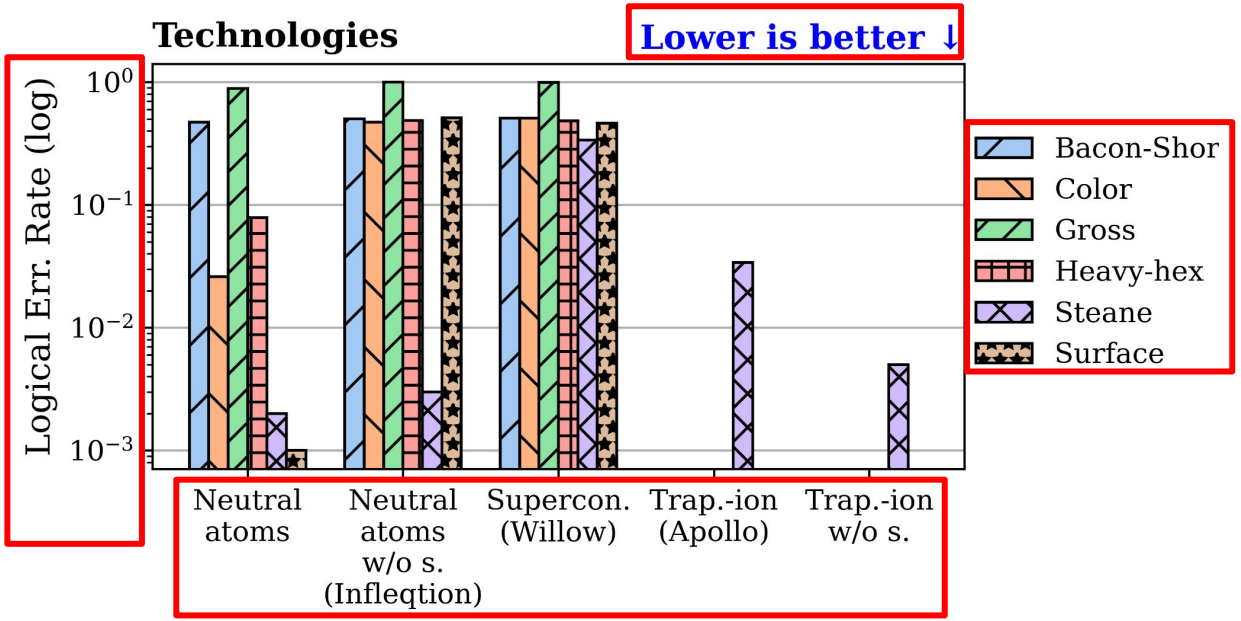
	Superconducting	Trapped-ion	Neutral atoms
Topology	Heavy-hex, grid	Race track, grid	2D array, arbitrary 3D geometry
Shuttling	No	Yes	Yes
Coherence time	10 - 100 µs	4 - 3600 s	1 - 60s
Operation speed	12 - 25 ns	1 - 10 µs	0.4 - 2 µs

Q2: Technology context

	Superconducting	Trapped-ion	Neutral atoms
Topology	Heavy-hex, grid	Race track, grid	2D array, arbitrary 3D geometry
Shuttling	No	Yes	Yes
Coherence time	10 - 100 µs	4 - 3600 s	1 - 60s
Operation speed	12 - 25 ns	1 - 10 µs	0.4 - 2 µs

Trapped-ion devices appear particularly promising due to their long coherence

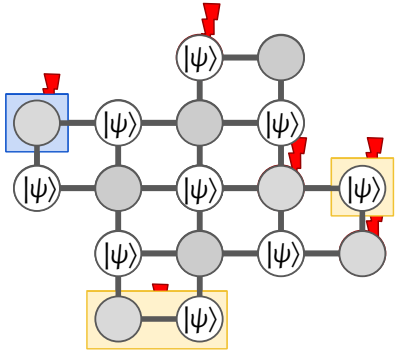
Q2: Technology results



Trapped-ion achieves the best overall reliability,
while shuttling reduces error rates by up to **49.8%** across backends

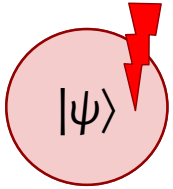
Q3: QEC importance context

Qubit protected with QEC:



- Decoherence errors
- Single- and two-qubit gate errors
- Readout errors
- Crosstalk errors

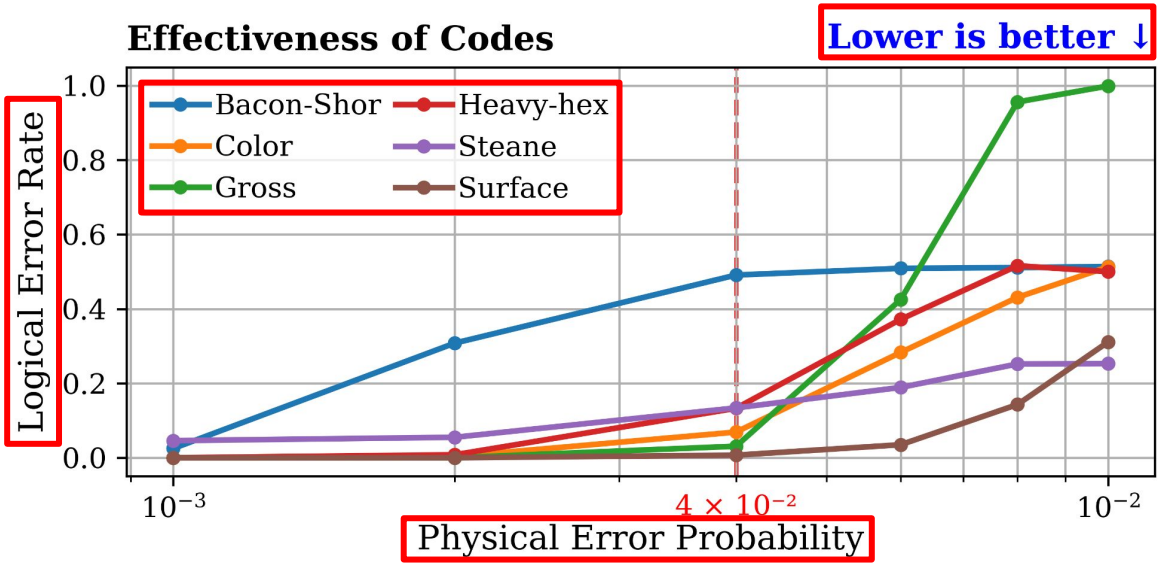
Unprotected qubit:



- Decoherence errors

QEC circuit may introduce notably more errors than unprotected qubit

Q3: QEC importance results



At **p = 0.004**, all codes exceed threshold while the unprotected qubit remains correct, showing QEC overhead outweighs its benefits at this error rate

Summary

We need QEC to make reliable quantum computation possible

How to evaluate suitability of QEC codes for practical applications?

- Missing classification of codes
- Intractable exploration space
- Fragmentation of the QEC toolchain

ECCentric: An Empirical Analysis of QEC Codes

Key takeaways

Trapped-ion hardware is the fastest path to reach fault tolerance

Logical errors are insensitive to qubit/readout quality variance

QEC is not always beneficial

Connectivity is more important than code distance

QEC-aware compilation is essential

Contact
Aleksandra Świerkowska
aleksandra.swierkowska@tum.de

Paper

Github