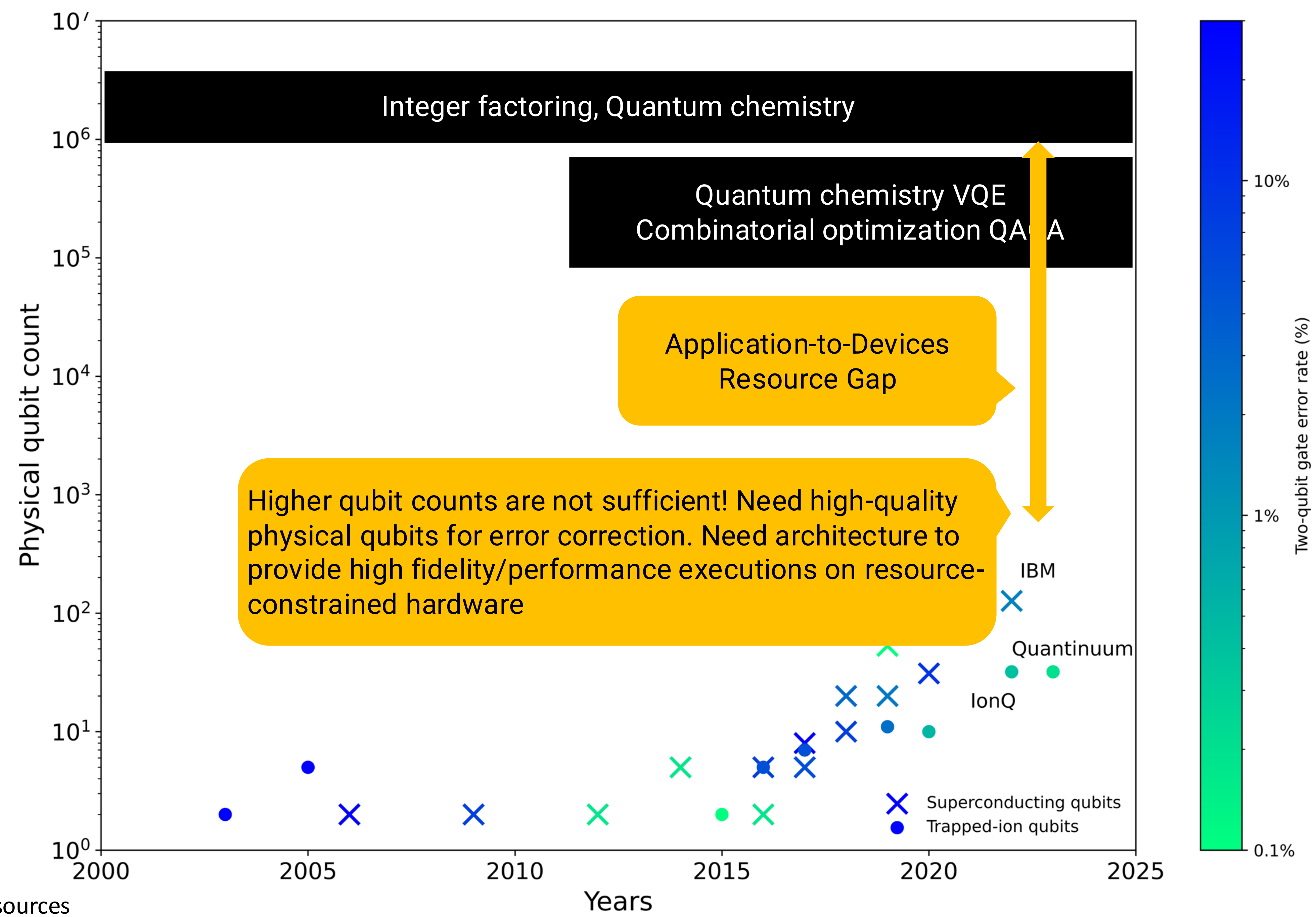
Architecting Scalable Quantum Computers Using Resource Estimation

Prakash Murali

Associate Professor of Computer Architecture

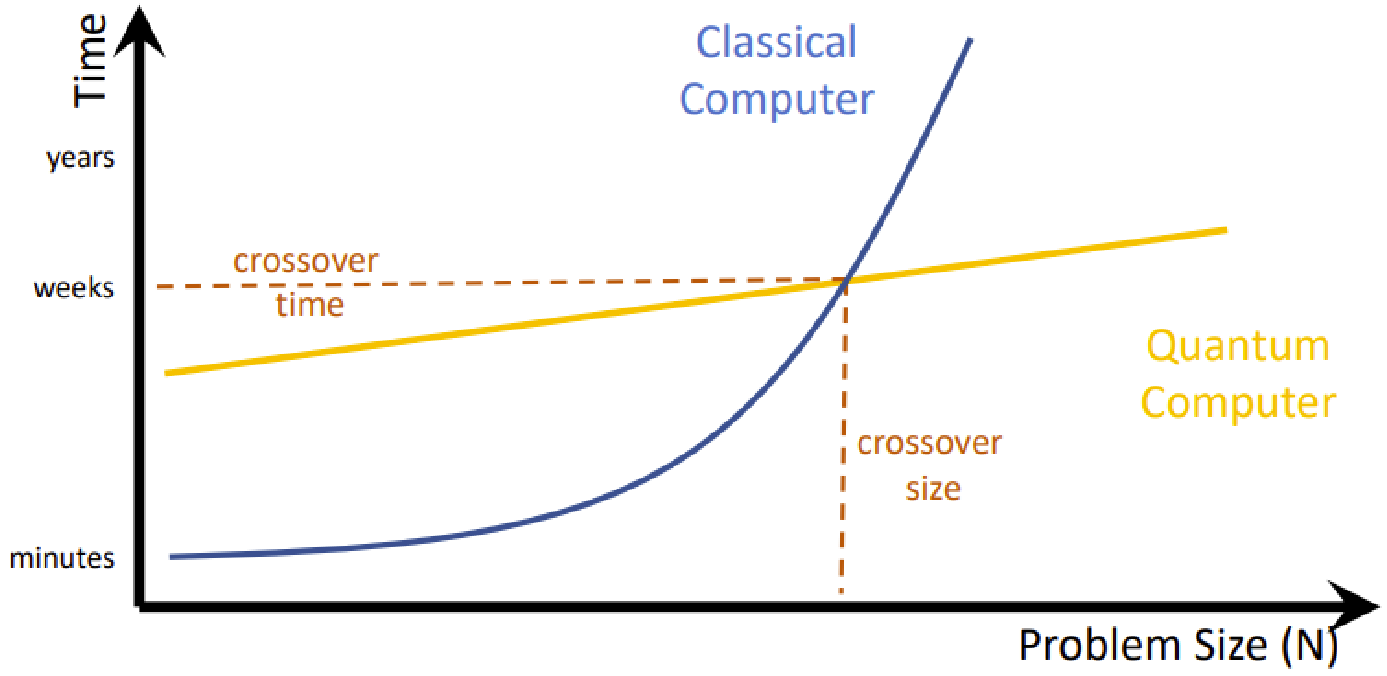
UNIVERSITY OF CAMBRIDGE



Data from multiple sources

Practically useful quantum computing

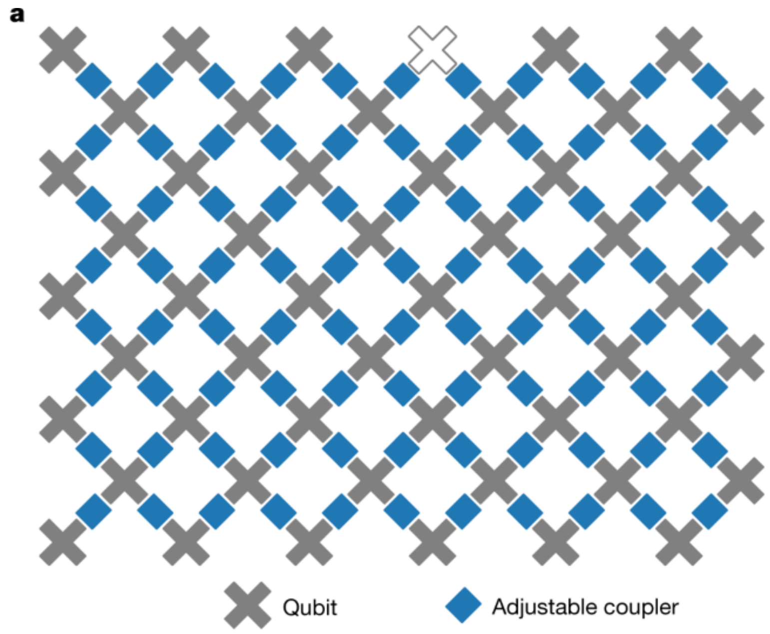
Need quantum applications with commercial or scientific relevance & reasonable resources needs (qubits, time)



Adapted from Hoefler et al. CACM 2023

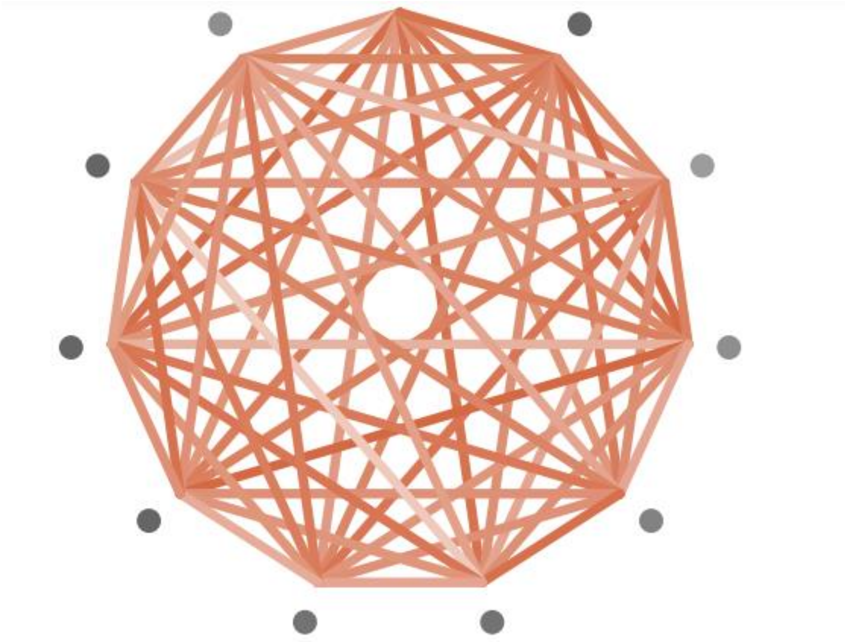
Challenge 1: A variety of hardware platforms

Superconducting qubits



Nanosecond operations, fSim/CZ gates

Trapped ion qubits



Microsecond operations, XX gates

Source: Google, IonQ

Challenge 2: A range of applications

Simplest scientific applications

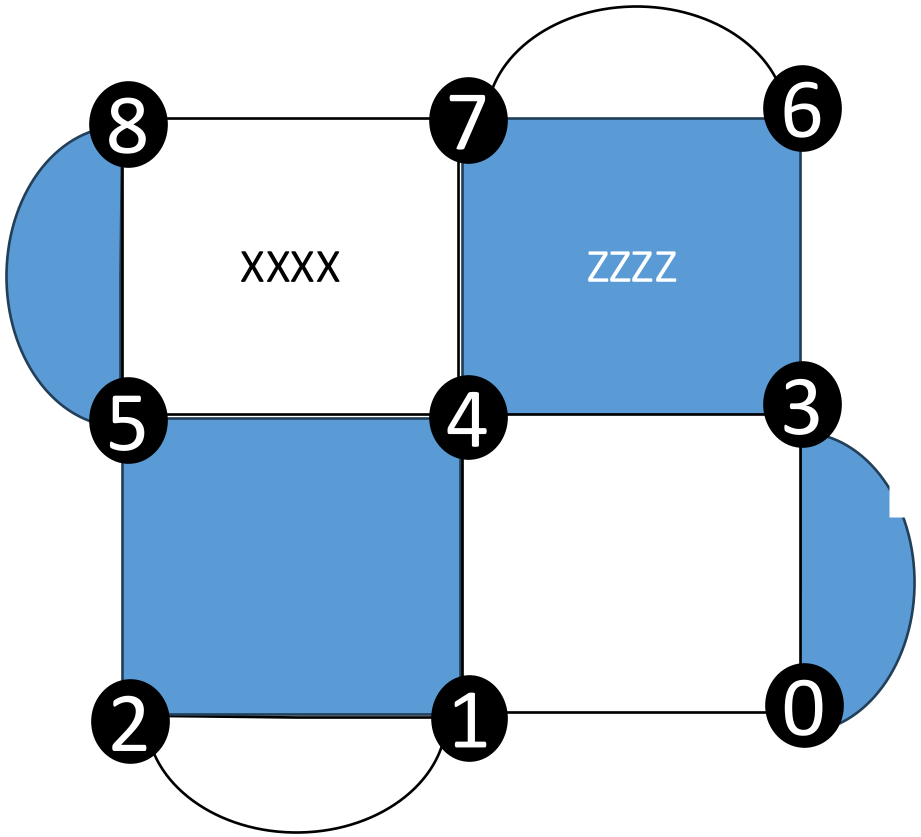
- 10x10 Ising model simulation
- 100 qubits
- 10000 operations`

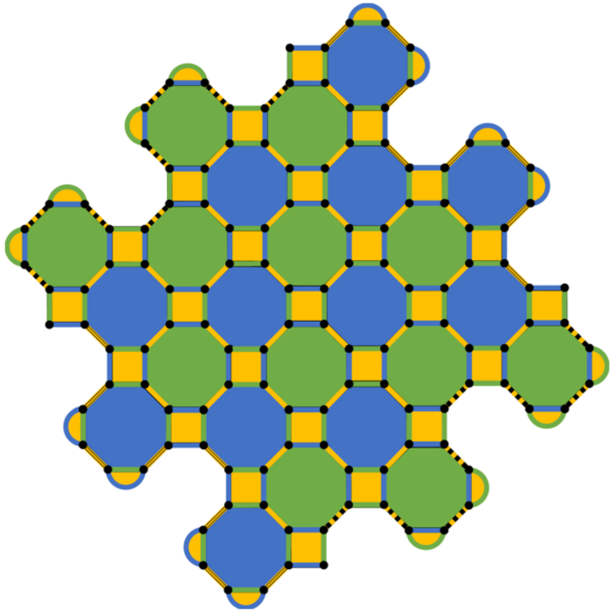
Large scale applications

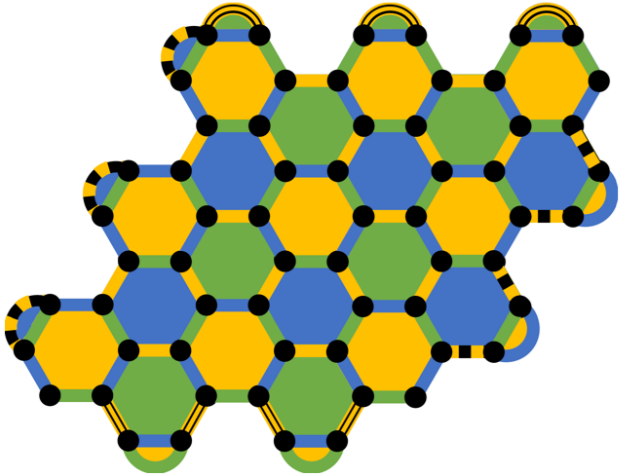
- 2048-bit RSA factoring
- 2000+ qubits
- 10^{10} operations

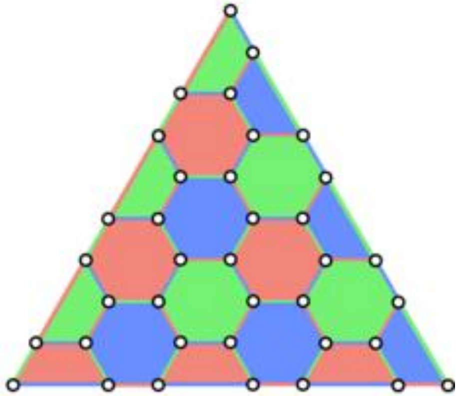
Challenge 3: A variety of error correction schemes

Surface code









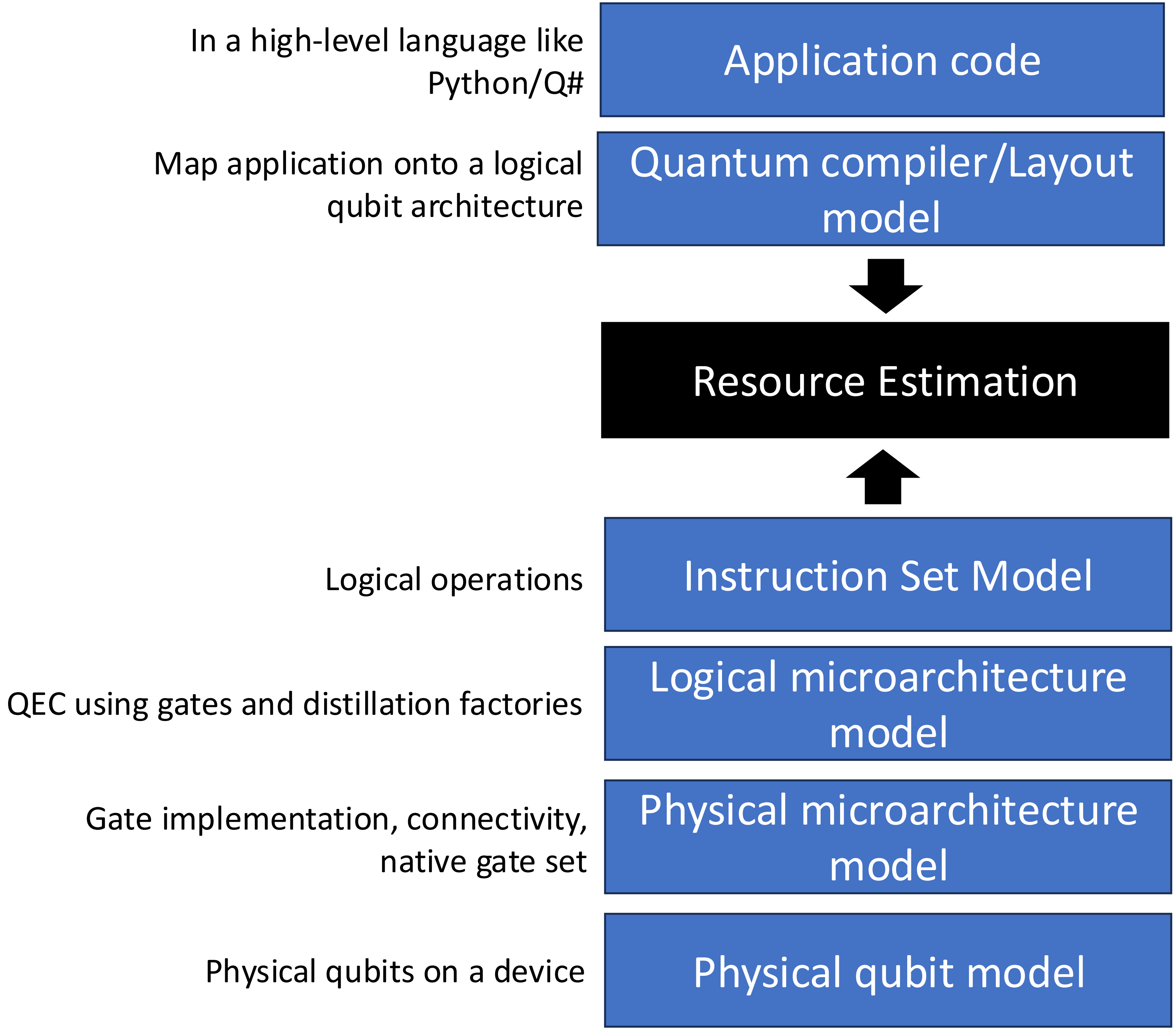
https://journals.aps.org/prxquantum/abstract/10.1103/PRXQuantum.5.030352

http://thequantuminsider.com/2022/05/12/microsoft-researchers-say-floquet-codes-boost-topological-qubit-error-correction/

Manual resource estimation is hard!

- Specific to one combination of algorithm and architecture
  - Factoring on superconducting qubits: Gidney & Ekera 2021
- Multiple interacting layers in the stack
- Focus on logical estimates:
  - Elliptic curve discrete logarithms Rotteler et al. 2017
- Hinders design exploration:
  - Need deep full-stack expertise
  - Hard to play with assumptions -> recomputations across the stack
  - Error prone

Technique 1: An appropriate set of abstractions

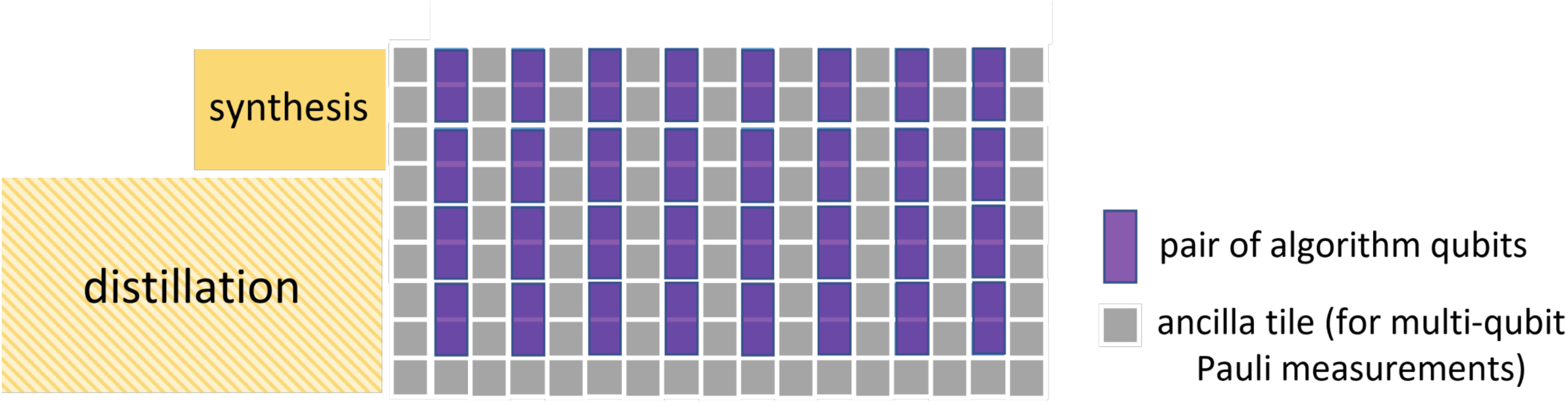


Technique 2: Scalable compiler & application modelling

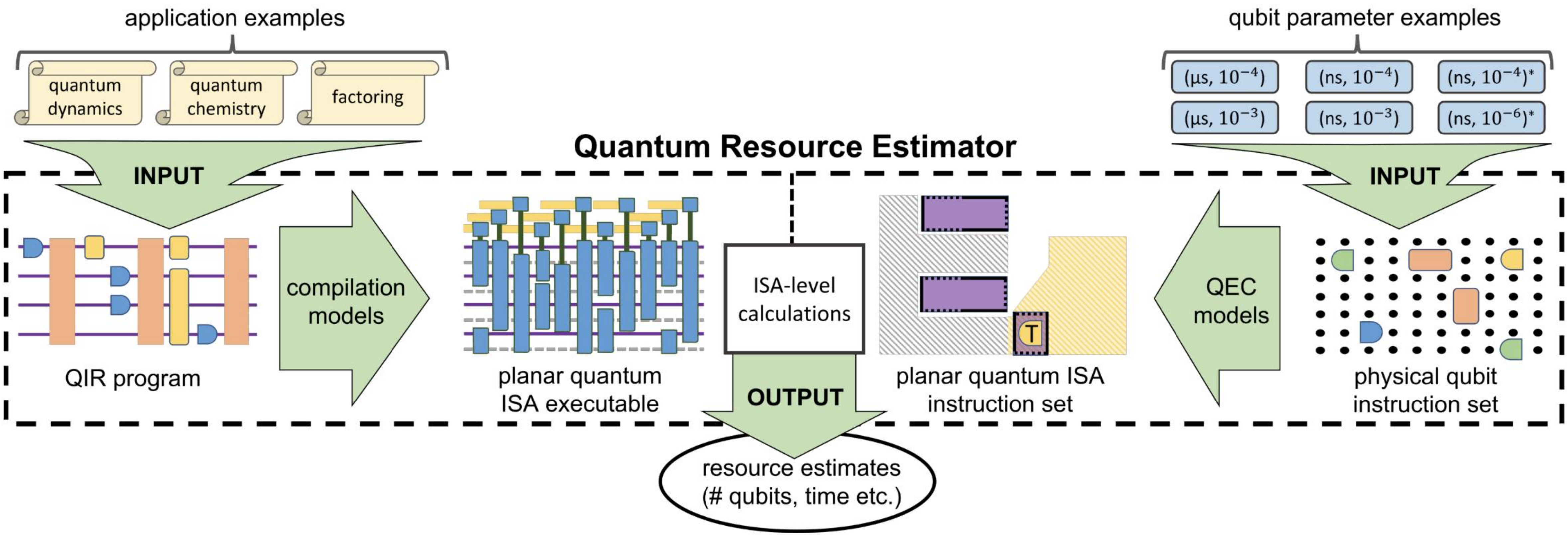
- Reduce application operations:
  - Use a compilation technique that removes Cliffords operations
  - Optimizations to parallelize rotation operations
- Accelerate parts needed for resource estimation:
  - Count operations rather than full-blown compilation
  - Caching of resource counts for functions and loops
  - User annotations in the program to help the compiler recognize parts where resource are the same

Technique 3: Automatic architecture optimization

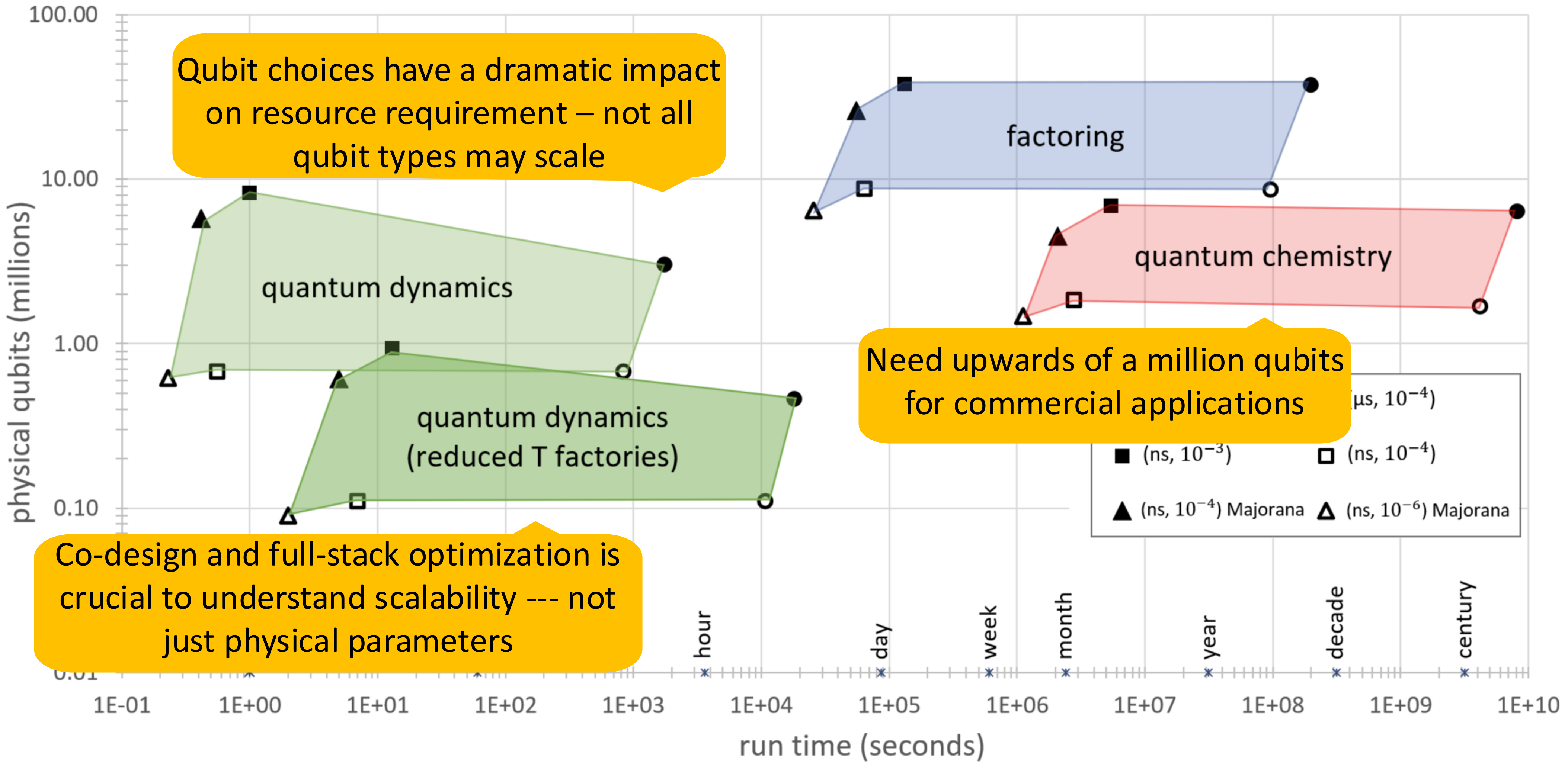
- Choose error correction properties considering physical qubit properties and application needs
- Choose the appropriate type and number of magic state factories
- Adjust magic state production vs. consumption needs



Resource estimation to guide architectural design



Co-design is critical for practical-scale quantum



Impact

- Informs scaling criteria for qubits:
    - **Fast:** Nanosecond operation speeds are beneficial to solve practical applications in under a month
    - **Reliable**: Physical operations with 10^{-4} or lower error rates
    - **Controllable**: Parallel operations across hundreds of thousands to million qubits

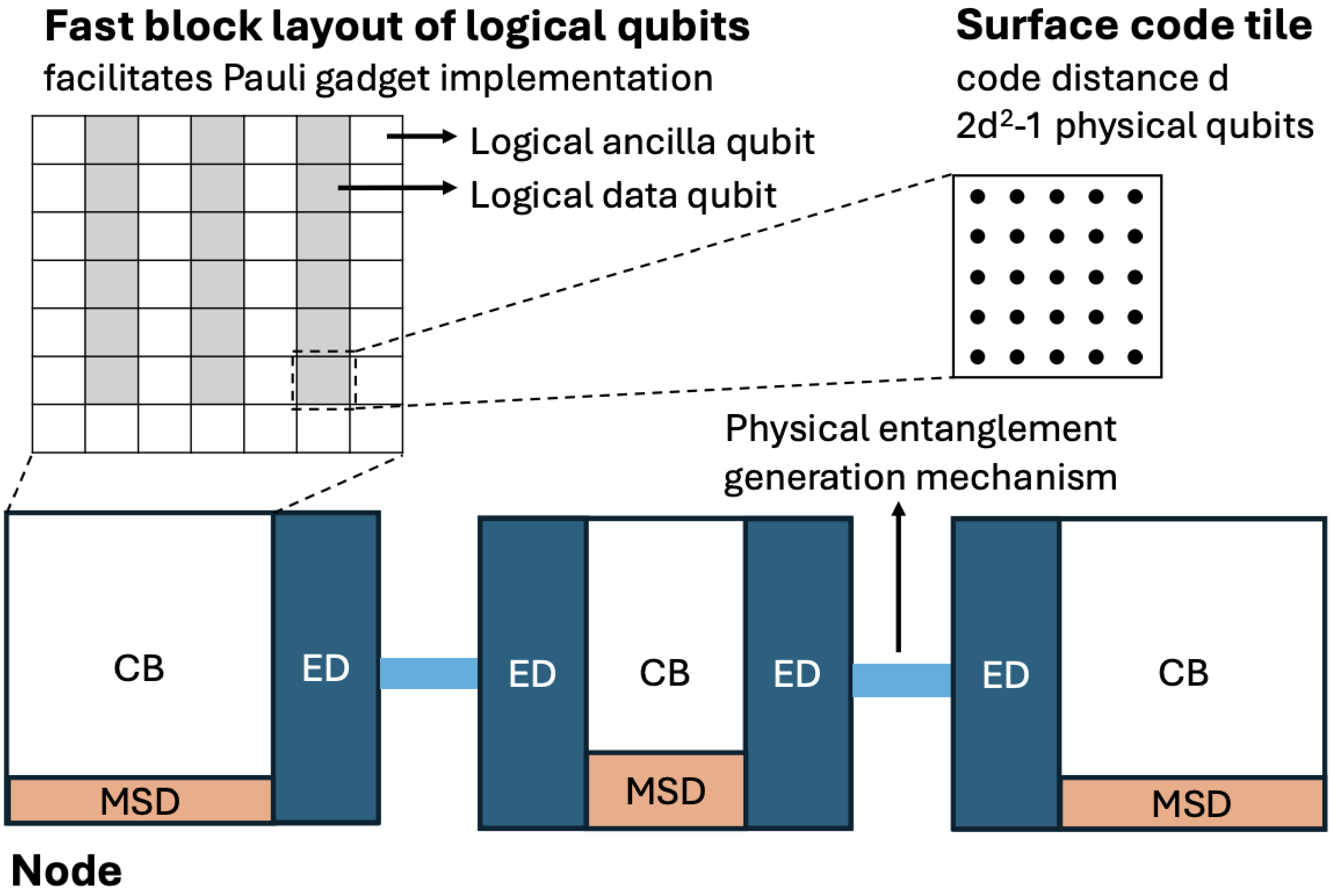
- Central to Microsoft's quantum strategy
- Spawned similar efforts from Google, Zapata and helps transition the community from noisy to fault-tolerant quantum

Need for distributed quantum computing

- Impractical monolithic device size:
  - Superconducting qubits ~1mm^2 or more per qubit. Million qubit devices need more than 1m^2 wafer
- Control challenges:
  - Multiple control wires per qubit. Very large wire counts and heating
- Imperfect yield:
  - Even at 100-qubit scales yield is poor. Chiplets offer a solution, but stills suffer yield challenges [Smith et al. MICRO’22]

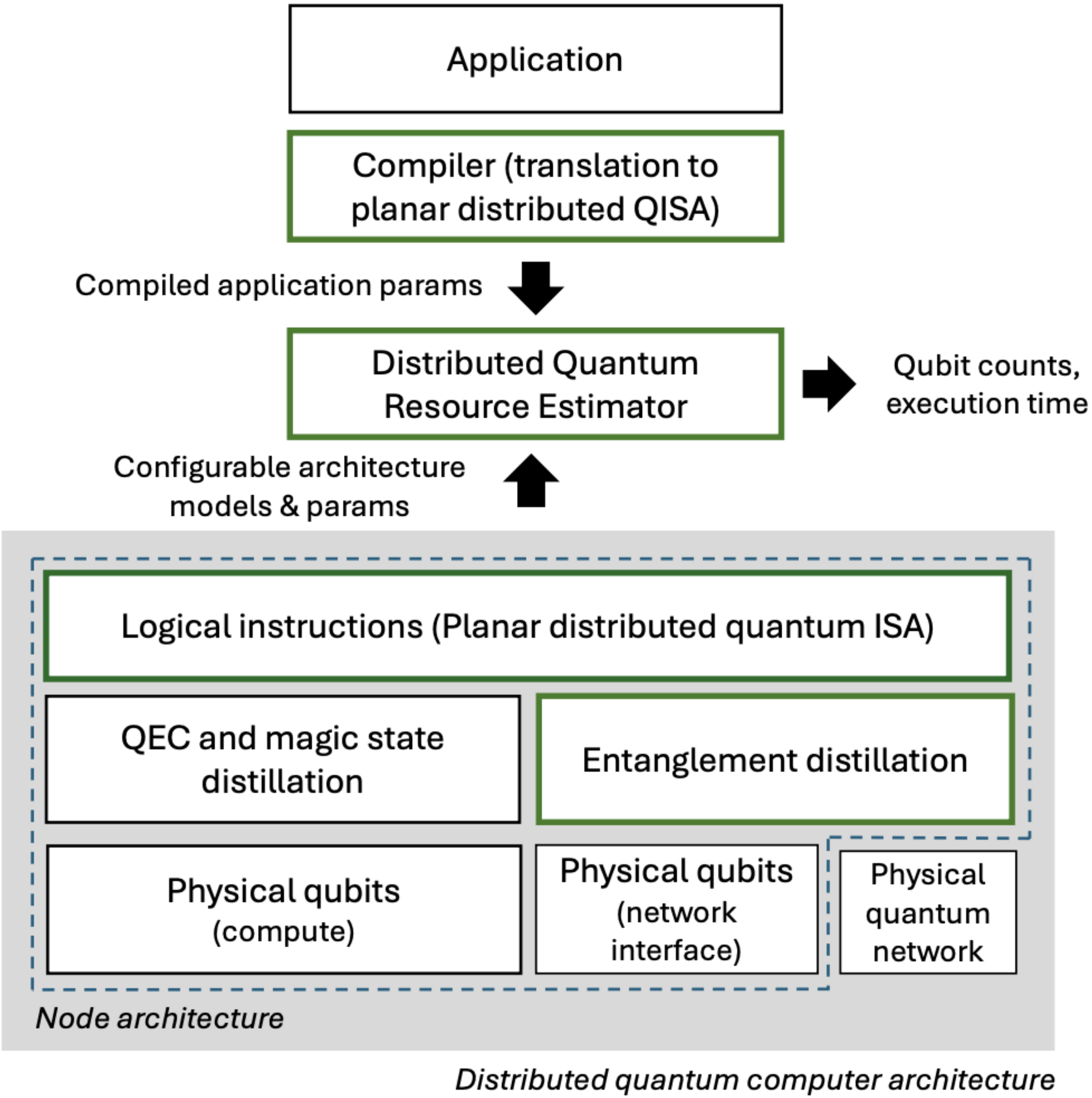
Architecture for distributed quantum computers

arXiv:2508.19160


Fast block layout of logical qubits
facilitates Pauli gadget implementation
Logical ancilla qubit
Logical data qubit
Surface code tile
code distance d
2d²-1 physical qubits
Physical entanglement
generation mechanism
CB
ED
MSD
ED
CB
ED
MSD
ED
CB
MSD
Node

Research questions

- What is a feasible architecture for a distributed quantum computer that can offer resource-efficient executions of practical-scale quantum applications?
- How should individual compute nodes be organised in terms of the quantum network, magic state distillation and logical qubits?
- What node sizes lead to overall resource-efficient executions?
- What are the costs of entanglement distillation?


Application
Compiler (translation to planar distributed QISA)
Compiled application params
Distributed Quantum Resource Estimator
Qubit counts, execution time
Configurable architecture models & params
Logical instructions (Planar distributed quantum ISA)
QEC and magic state distillation
Entanglement distillation
Physical qubits (compute)
Physical qubits (network interface)
Physical quantum network
Node architecture
Distributed quantum computer architecture

Distributed quantum computers are feasible

- 50ns gates, error rate of 10-4, 10MHz entanglement generation
- Spacetime requirement is 3-4X compared to monolithic architectures, but with only node sizes of 40-60K qubits (not 1M qubits on the same chip)

Application	Monolithic - Azure		Monolithic - Ours		Distributed - 1%	
	Qubits	Runtime	Qubits	Runtime	Qubits	Runtime
Ising 10x10	0.1111M	7.92 sec	0.0913M	7.92 sec	0.0881M	12.5 sec
Fermi-Hubbard 10x10	0.233M	51.5 min	0.260M	51.5 min	0.395M	1.59 hr
Heisenberg 10x10	0.181M	1.33 days	0.235M	1.34 days	0.314M	2.39 days
Shor's Factoring 2048	11.6M	18.8 hr	8.67M	16.3 hours	20.9M	1.25 days
ZnS QPE	0.367M	3.19 days	0.450M	3.22 days	0.941M	6.40 days
Benzene QPE	0.892M	16.7 days	0.750M	16.9 days	1.69M	29.8 days
Ruthenium QPE	1.86M	15.9 days	1.71M	15.9 days	2.31M	1.88 months
Nitrogenase QPE	2.41M	1.56 years	2.28M	1.56 years	3.53M	5.50 years

Insights on distributed architecture

- 30-60% of available qubits need to dedicated to entanglement distillation
- Speed matching is important – slow qubit types can tolerant low entanglement generation rates, easing network requirements
- 1% Bell state error rate is a good goal for future hardware Current targets of 0.1% are not required*
- Detailed analysis in the paper on node sizing, error rates etc. (arXiv:2508.19160)

*https://defencescienceinstitute.com/wp-content/uploads/2025/ 07/DARPA-SN-25-98.pdf

Takeaways

1. Large gap between application needs and hardware capability
2. Resource estimation helps us model and design for practical quantum advantage
3. Large devices beyond 1M qubits are needed for practical quantum advantage
4. Distributed quantum computers offer a feasible path forward. Need 3-4X resources of monolithic designs, but allow us to limit device sizes to range of 40-60K qubits

prakashmurali.bitbucket.io