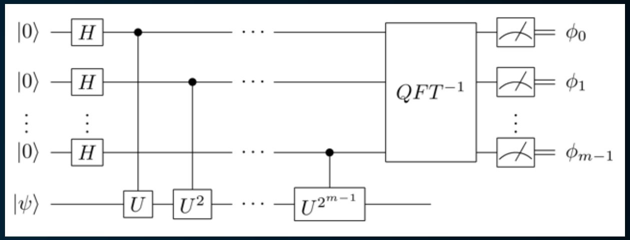
Phase estimation variants

and its implication for quantum/classical architecture

Outline

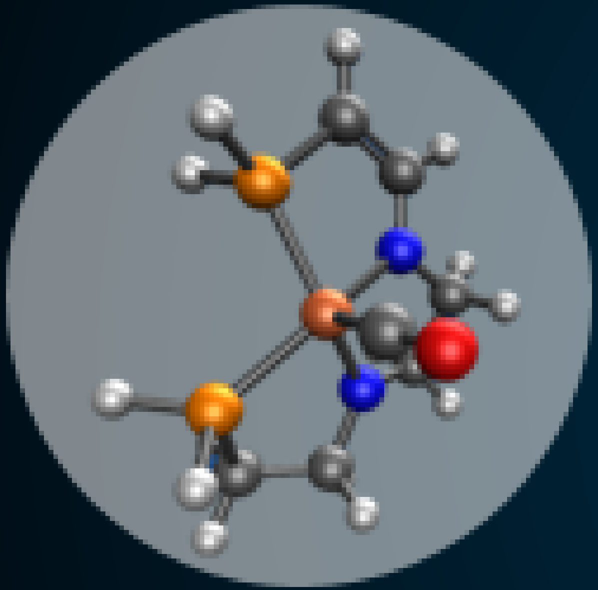
- Phase estimation variants and their architecture requirements



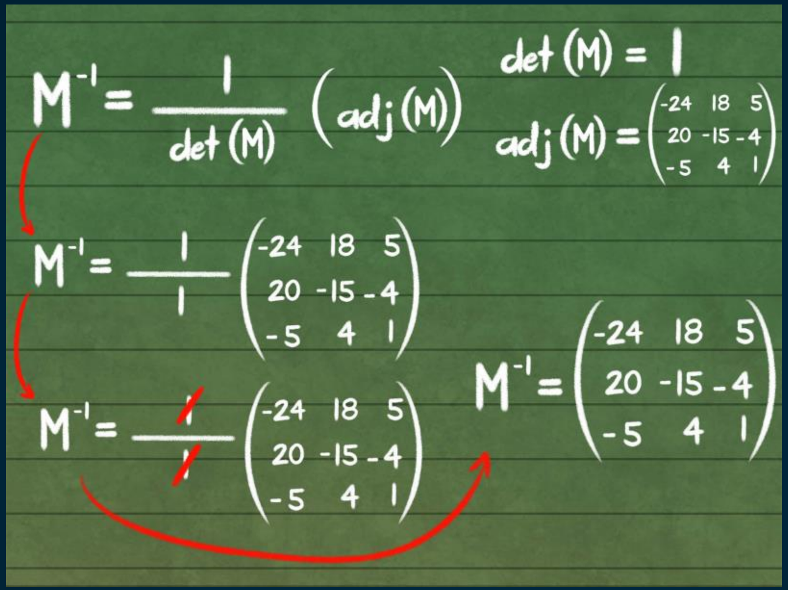
- 5 levels of quantum / classical integration

- The case for a hybrid Intermediate Representation

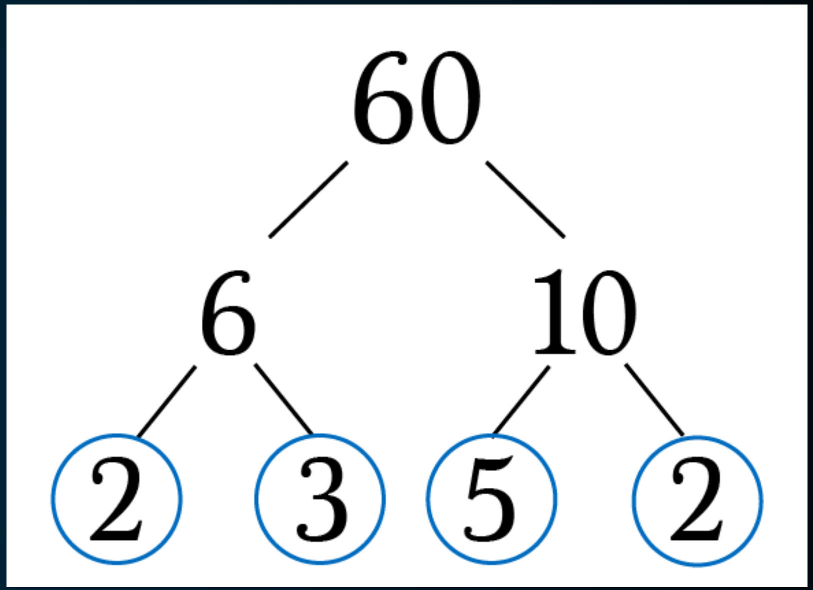
Quantum phase estimation: use cases



Molecular simulation
and
Material design

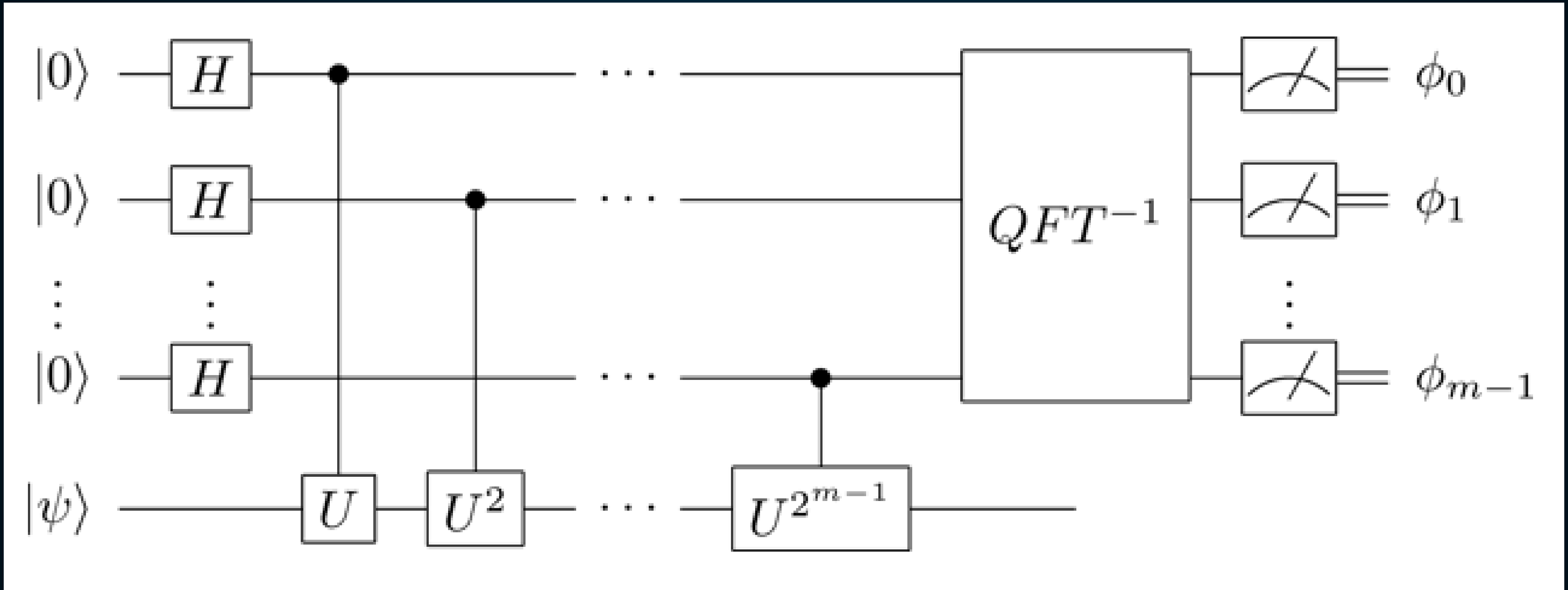


Linear system solving

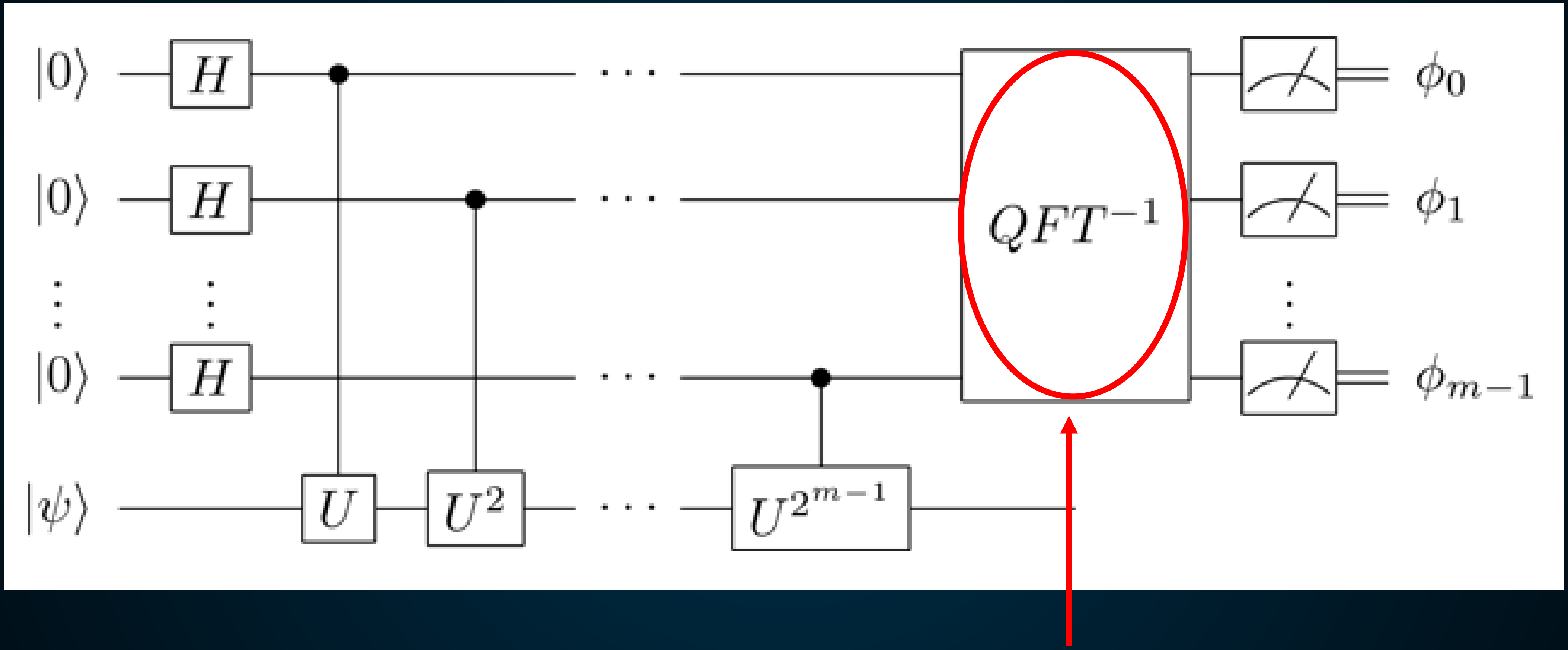


Number factoring

Quantum phase estimation: circuit

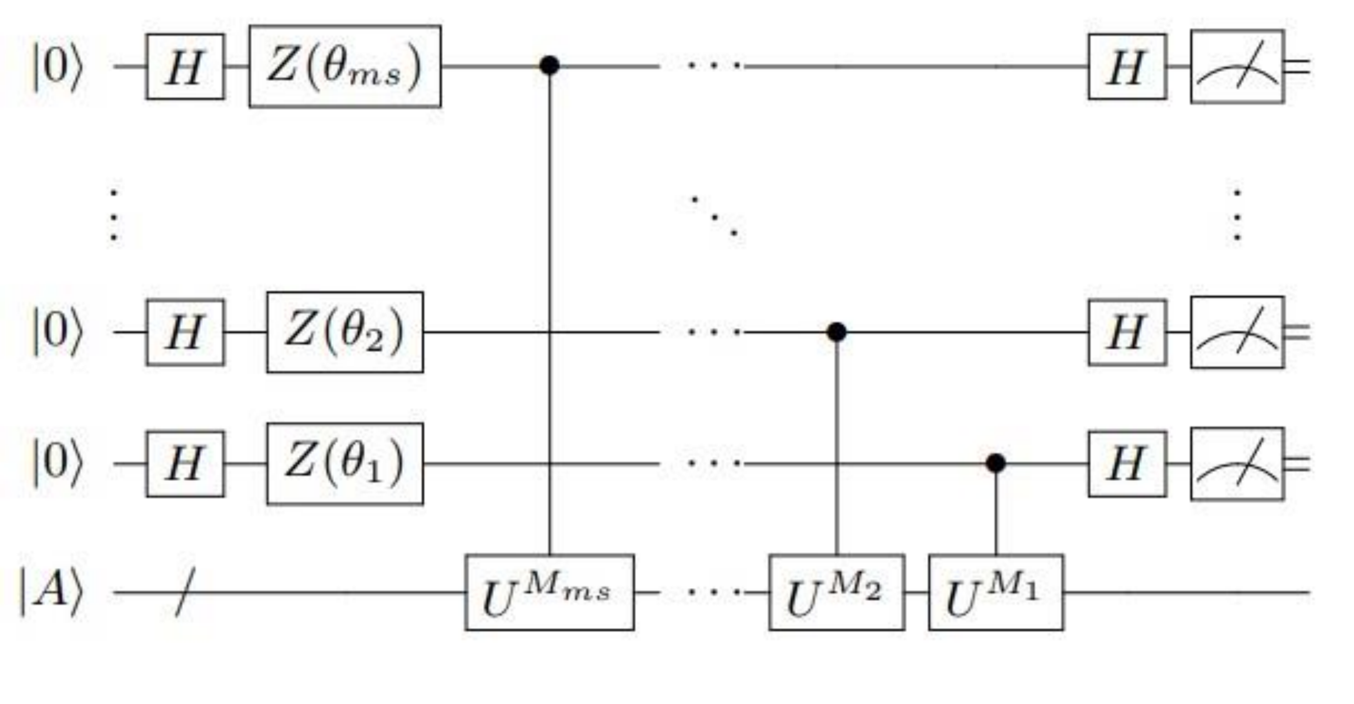


Quantum phase estimation

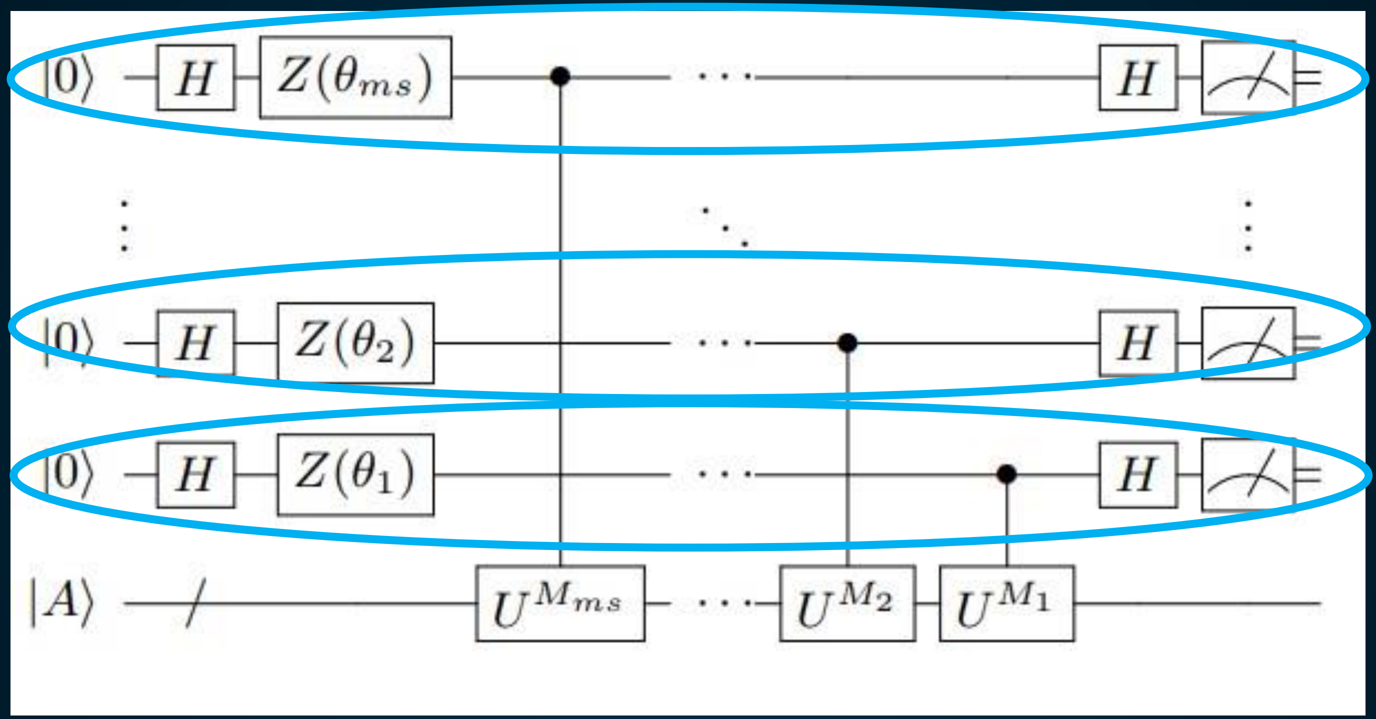


Auxiliary qubits are coupled by the inverse Fourier Transform.

Iterative phase estimation

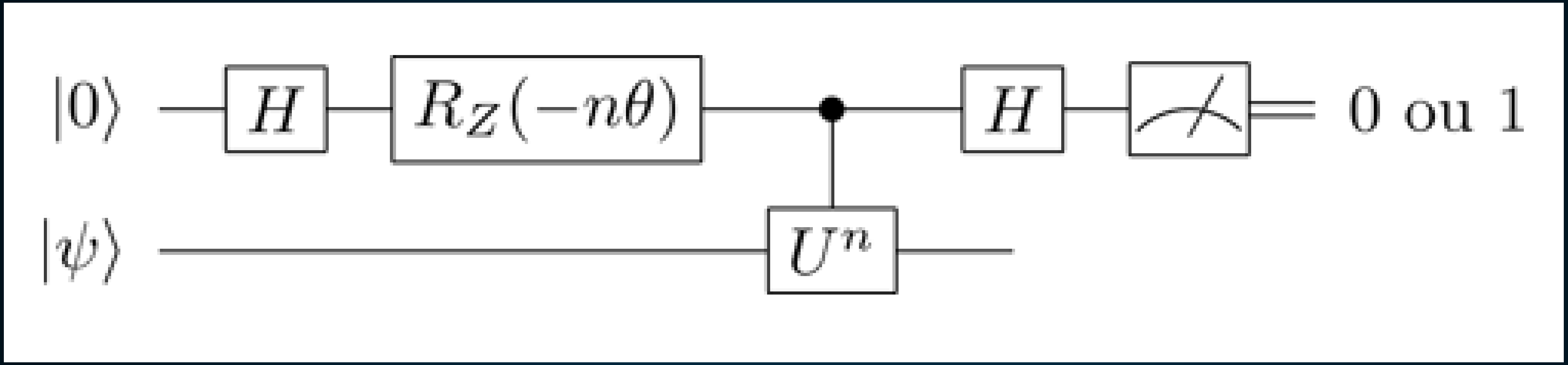


Iterative phase estimation



- Replacing the Fourier transform by a Hadamard transform decouples auxiliary qubits and opens the door to parallelization.
- But using a Hadamard transform requires classical postprocessing to compute the estimated phase.

Iterative phase estimation



- Smaller circuits:
  - only one auxiliary qubit.
  - reduced depth.

-> lower requirement for quantum hardware.

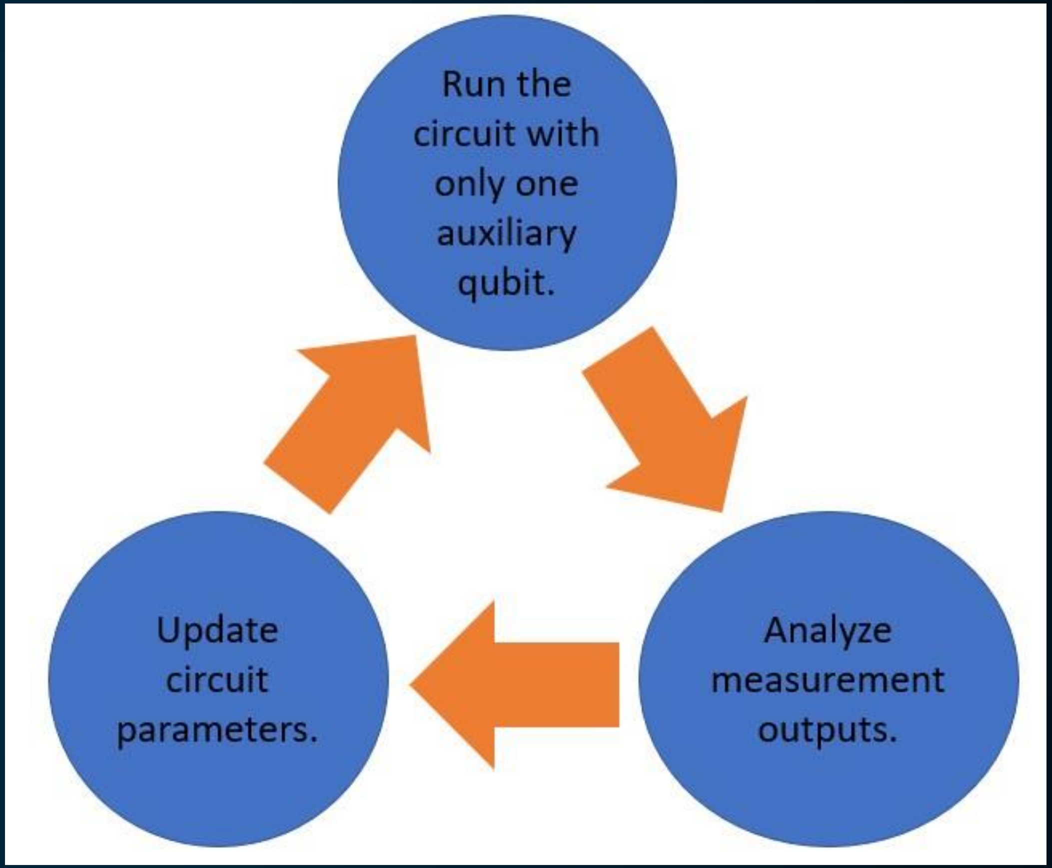
$$\Pr(0) = \cos^2\left(n\frac{\phi-\theta}{2}\right).$$

$$\Pr(1) = \sin^2\left(n\frac{\phi-\theta}{2}\right).$$

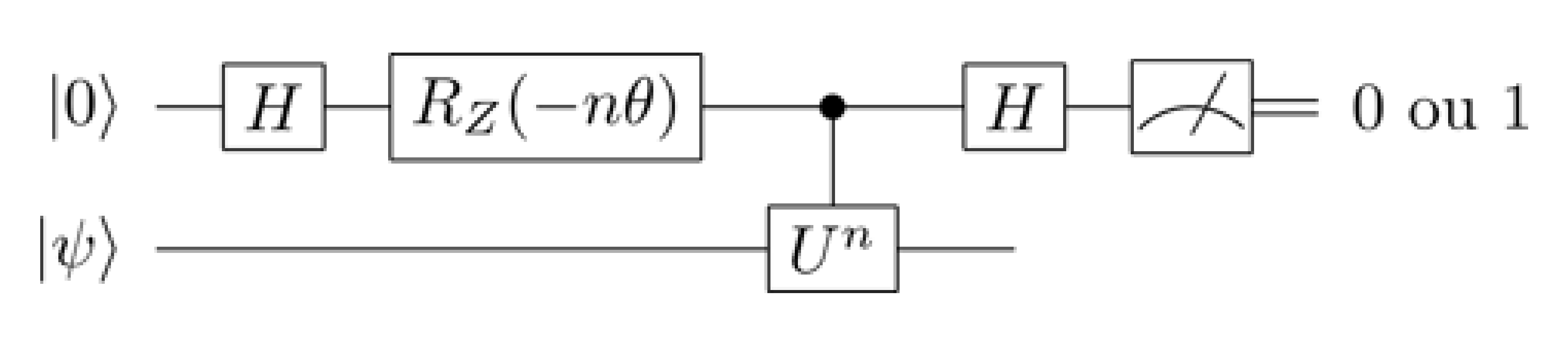
Iterative phase estimation

- Smaller circuits:
    - only one auxiliary qubit.
    - reduced depth.

-> lower requirement for quantum hardware.



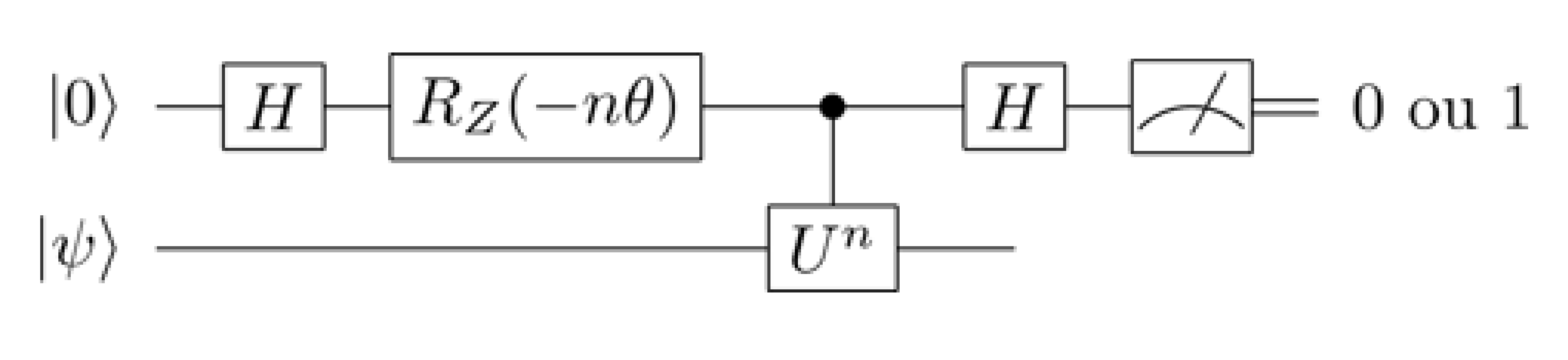
Iterative phase estimation



How should we choose the sequence of values for θ and n ?

-> different tradeoffs lead to different flavors of iterative phase estimation.

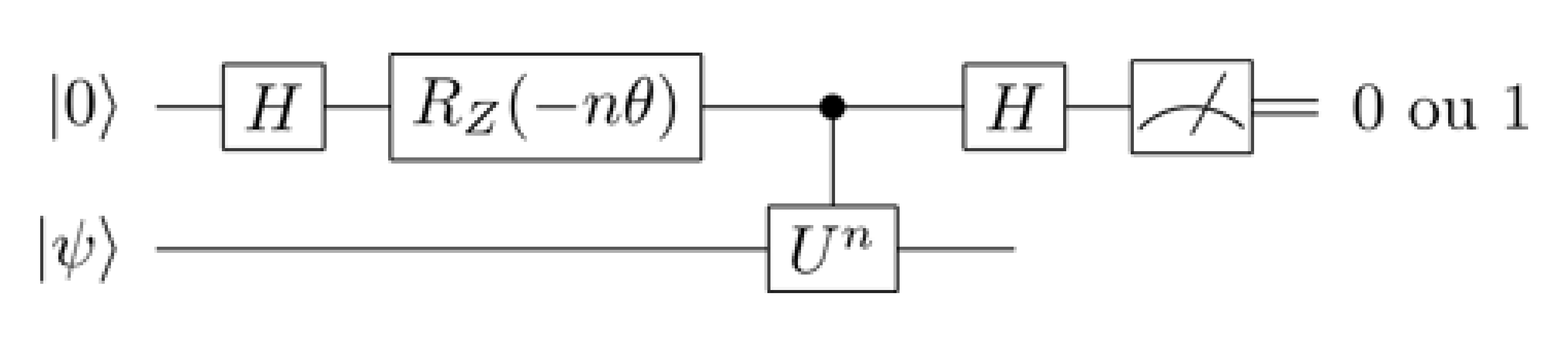
Bayesian phase estimation



- For m bits of precision for the estimated phase ϕ:
  - 2^m classical values are considered.
  - Updating our current knowledge of ϕ requires O(2^m) classical computation.
  - Optimal in the number of queries to U.

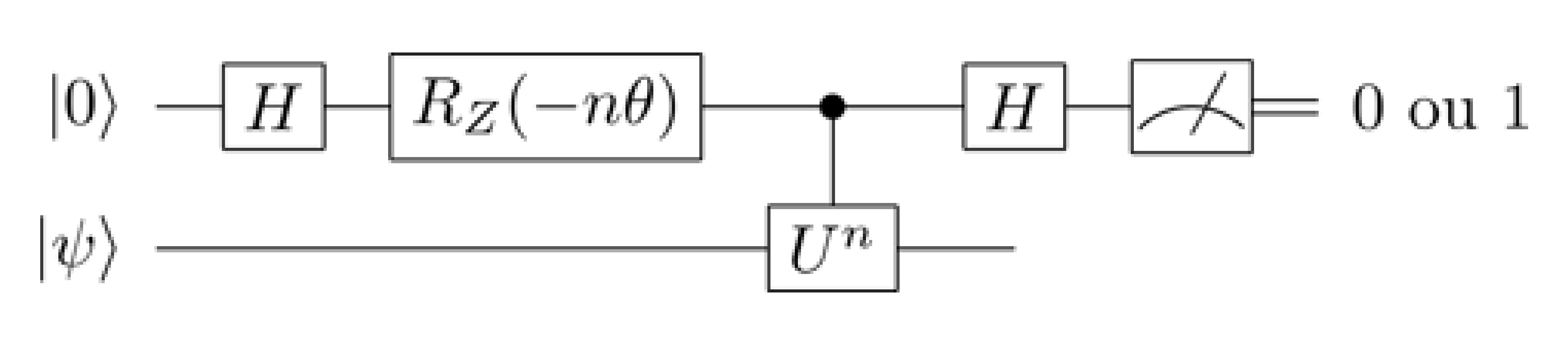
$$\Pr(\phi_a|d) = \frac{\Pr(d|\phi_a)\Pr(\phi_a)}{\Pr(d)}.$$

Robust phase estimation



- Efficient in terms of classical computing.
- More measurements to learn most significant bits of ϕ than to learn least significant bits.
- Robust to state preparation and measurement (SPAM) errors.
- Precision in ϕ scales like $\frac{c}{Q}$. (optimal scaling is $\frac{d}{Q}$ with $d < c$. Q is the number of queries to U.)

Random walk phase estimation



- Estimated ϕ is modeled by a Gaussian distribution (2 parameters only).
- Efficient scaling of the precision of ϕ with respect to:
  - Queries to U.
  - Classical postprocessing.
- Adaptive algorithm : subsequent values of n and θ depend on previous measurement outcomes.

-> requires close integration of quantum and classical computations.

Resources to go further

- Iterative and Bayesian phase estimation:
  - [1304.0741] Faster Phase Estimation (arxiv.org)
  - Quantum/BayesianPhaseEstimation.qs at main · microsoft/Quantum (github.com)
- Robust phase estimation:
  - RobustPhaseEstimation operation - Q# reference - Microsoft Quantum | Microsoft Docs
  - [1502.02677] Robust Calibration of a Universal Single-Qubit Gate-Set via Robust Phase Estimation (arxiv.org)
  - QuantumLibraries/Robust.qs at main · microsoft/QuantumLibraries (github.com)
- Iterative and Bayesian phase estimation:
  - [1309.0876] Hamiltonian Learning and Certification Using Quantum Resources (arxiv.org)
  - Quantum-NC/RandomWalkPhaseEstimation.qs at main · microsoft/Quantum-NC (github.com)
- Blog post (in French): Quantum Computing | Estimation de phase : forces et faiblesses des variantes (vivienlonde.github.io)

Hybrid Quantum Computing Architectures

1. Batch quantum computing

Shor

2. Interactive batch quantum computing

VQE

3. Integrated quantum computing

Adaptive phase estimation

4. High performance quantum computing

Rejection sampling

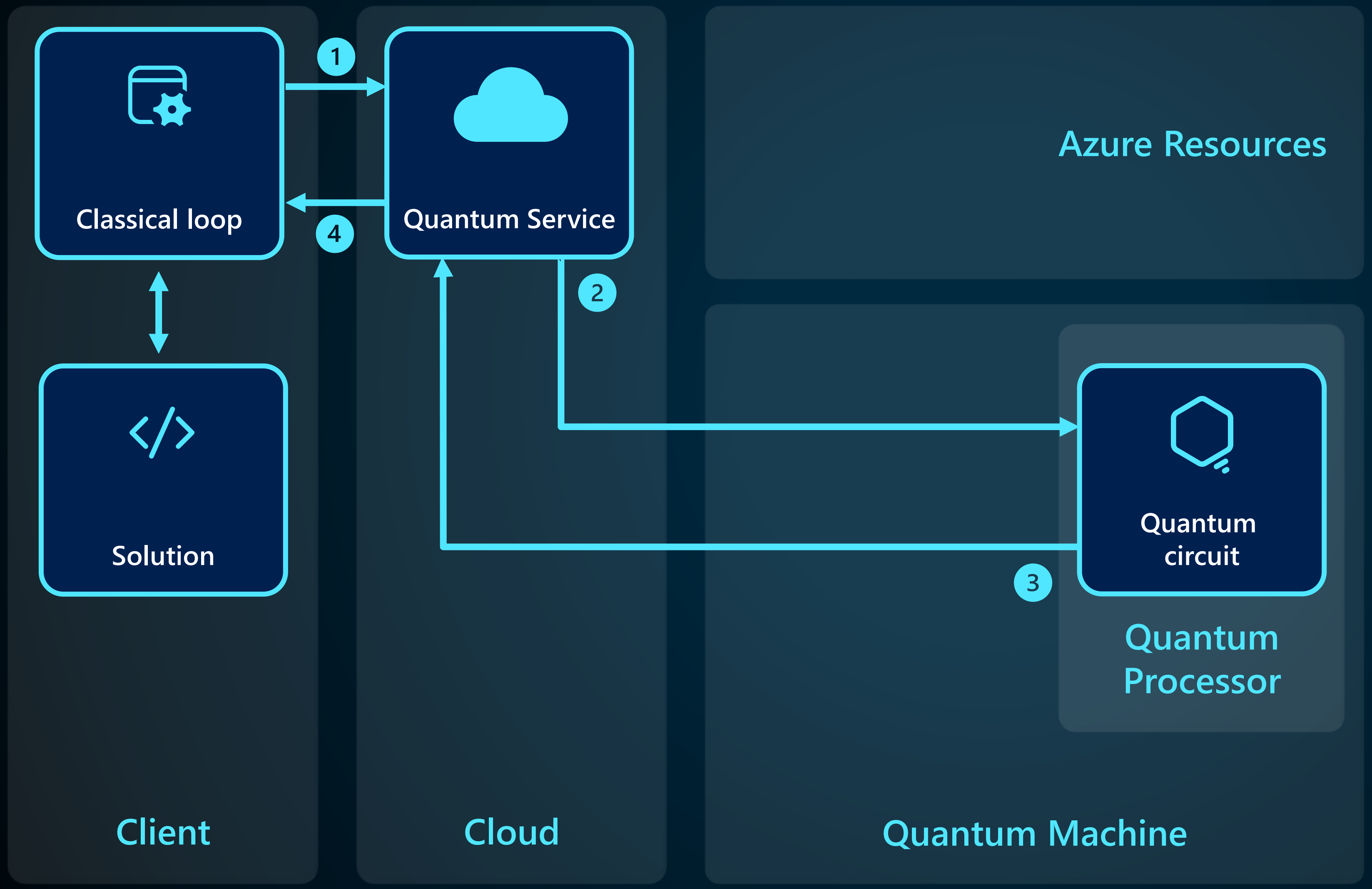
5. Distributed quantum computing

Materials modeling

Increasingly tighter and more complex interaction between quantum & classical processing

Level 1: Batch quantum computing

Quantum circuit with classical pre- and post-processing

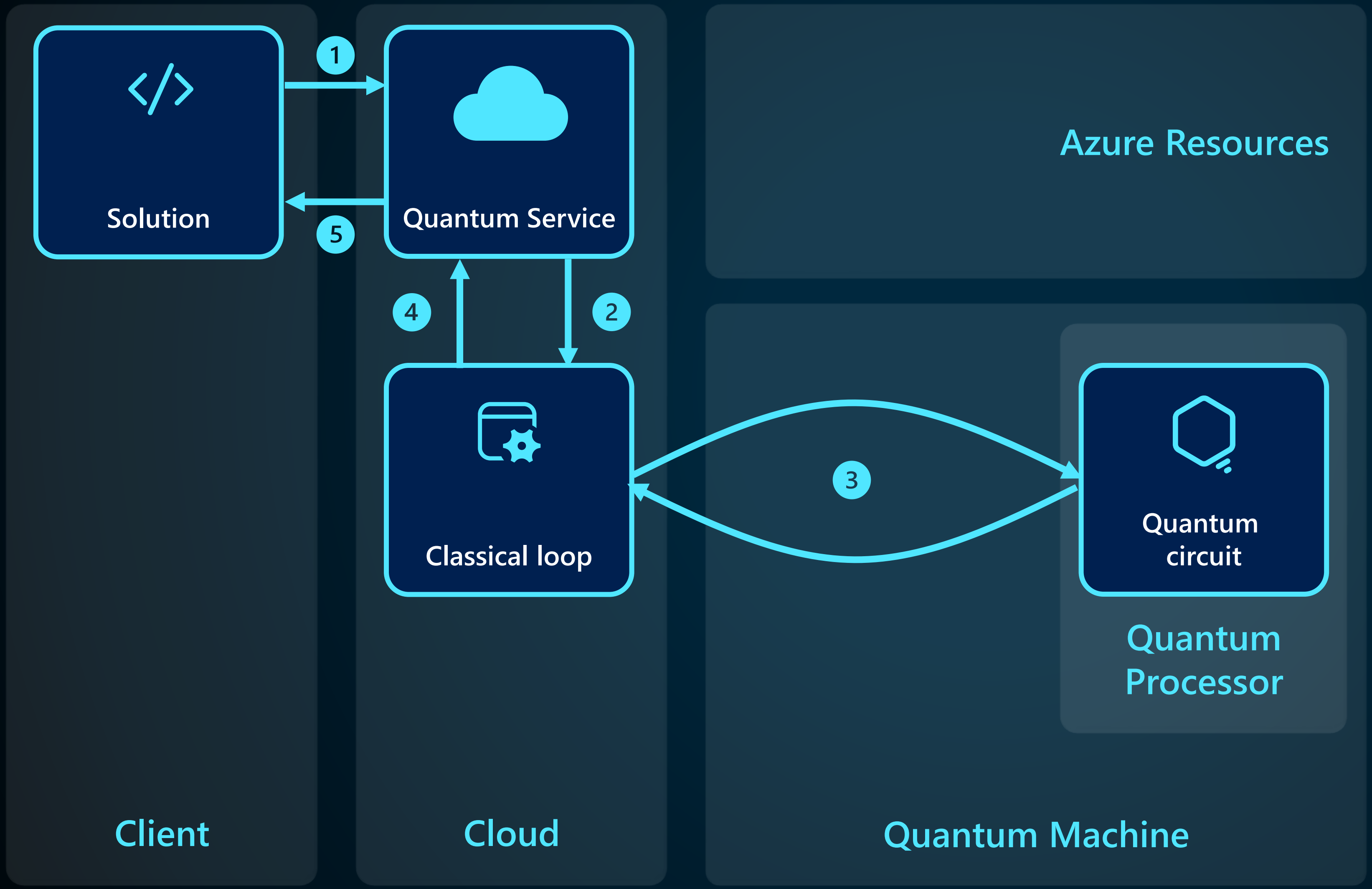


Examples

- Shor's algorithm (cryptanalysis)
- Simple quantum phase estimation

Level 2: Interactive batch quantum computing

Parameterized quantum circuit in a classical driver loop that runs in the cloud



(3) Prioritized loop

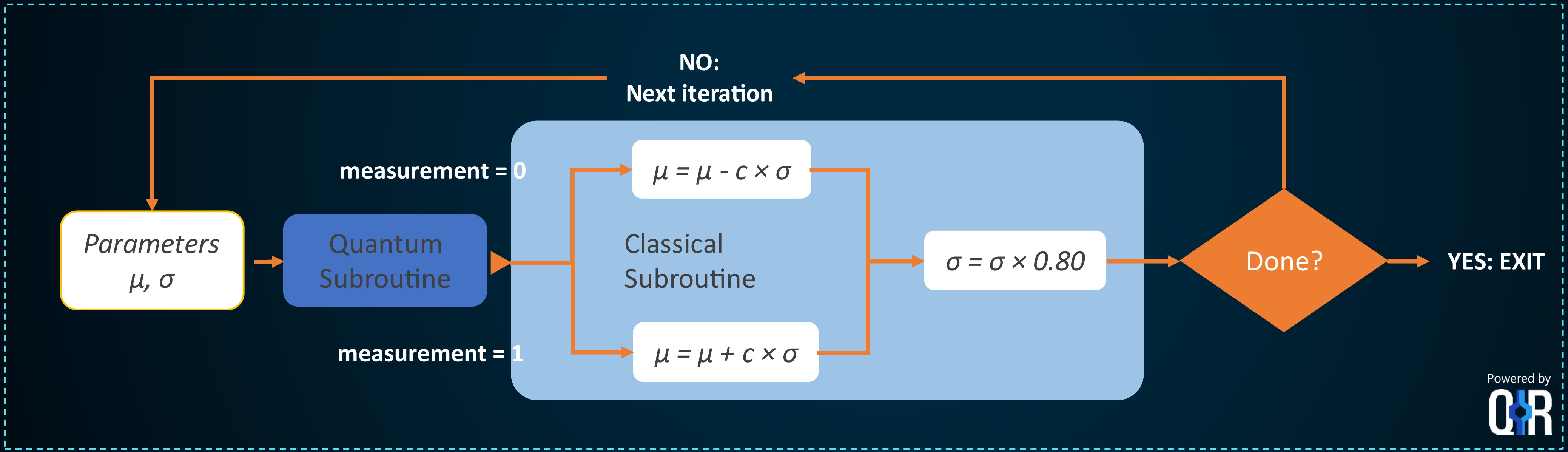
Examples

- Variational quantum eigensolvers (VQE)
- Quantum approximate optimization algorithms (QAOA)

3. Integrated quantum computing

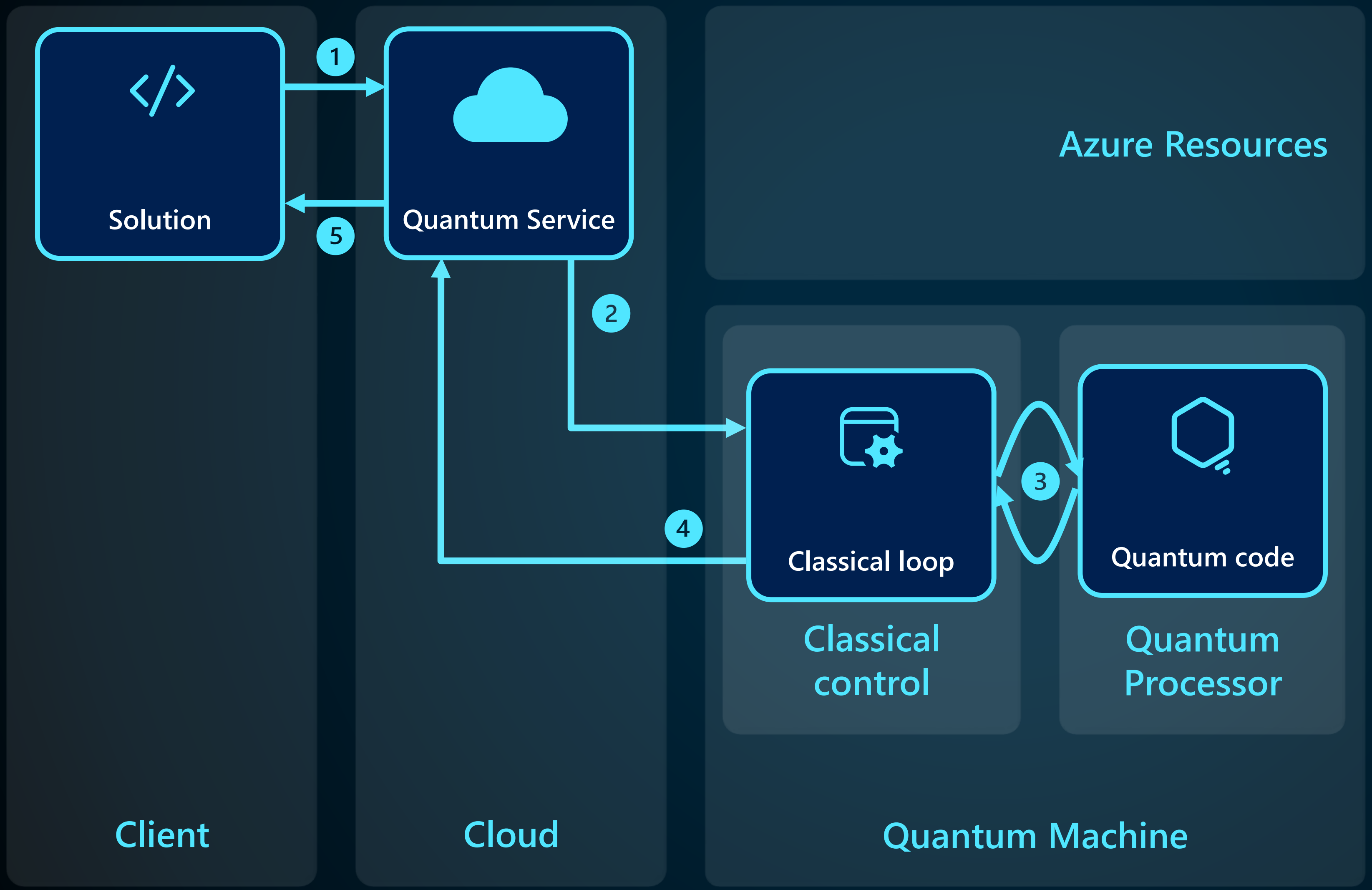
In the quantum machine:

- Arbitrary control flow,
- Classical computations by back-end while physical qubits are alive



Level 3: Integrated quantum computing

Coherent quantum coroutine with a classical driver loop



3 Physical qubit remains alive

Limited classical control

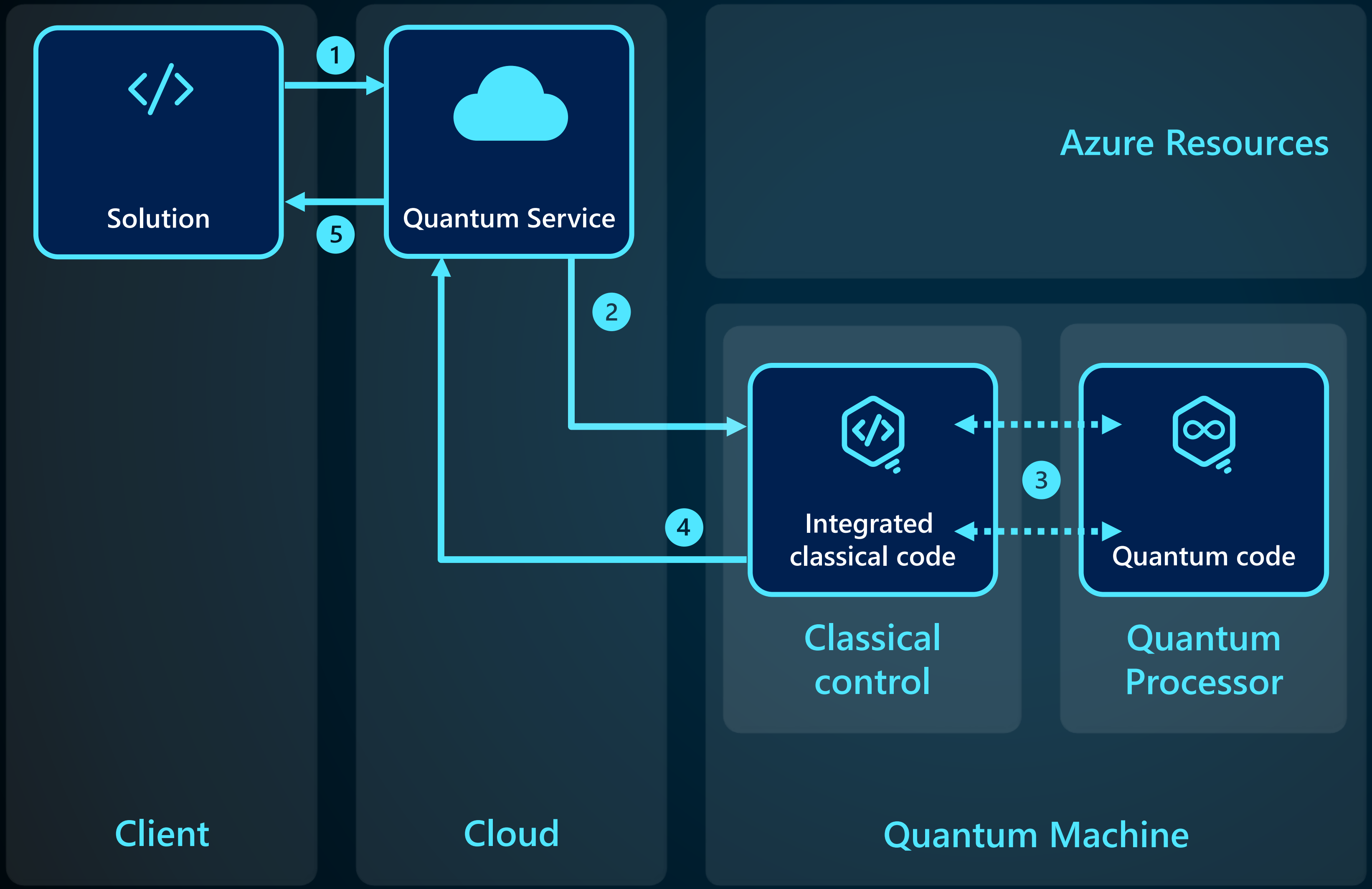
Example

Parameterized quantum coroutine:

- Adaptive phase estimation techniques such as random walk PE or Bayesian PE
- Error Correction

Level 4: High Performance quantum computing

Full classical compute next to QPU



3 Logical qubit with indefinite lifetime

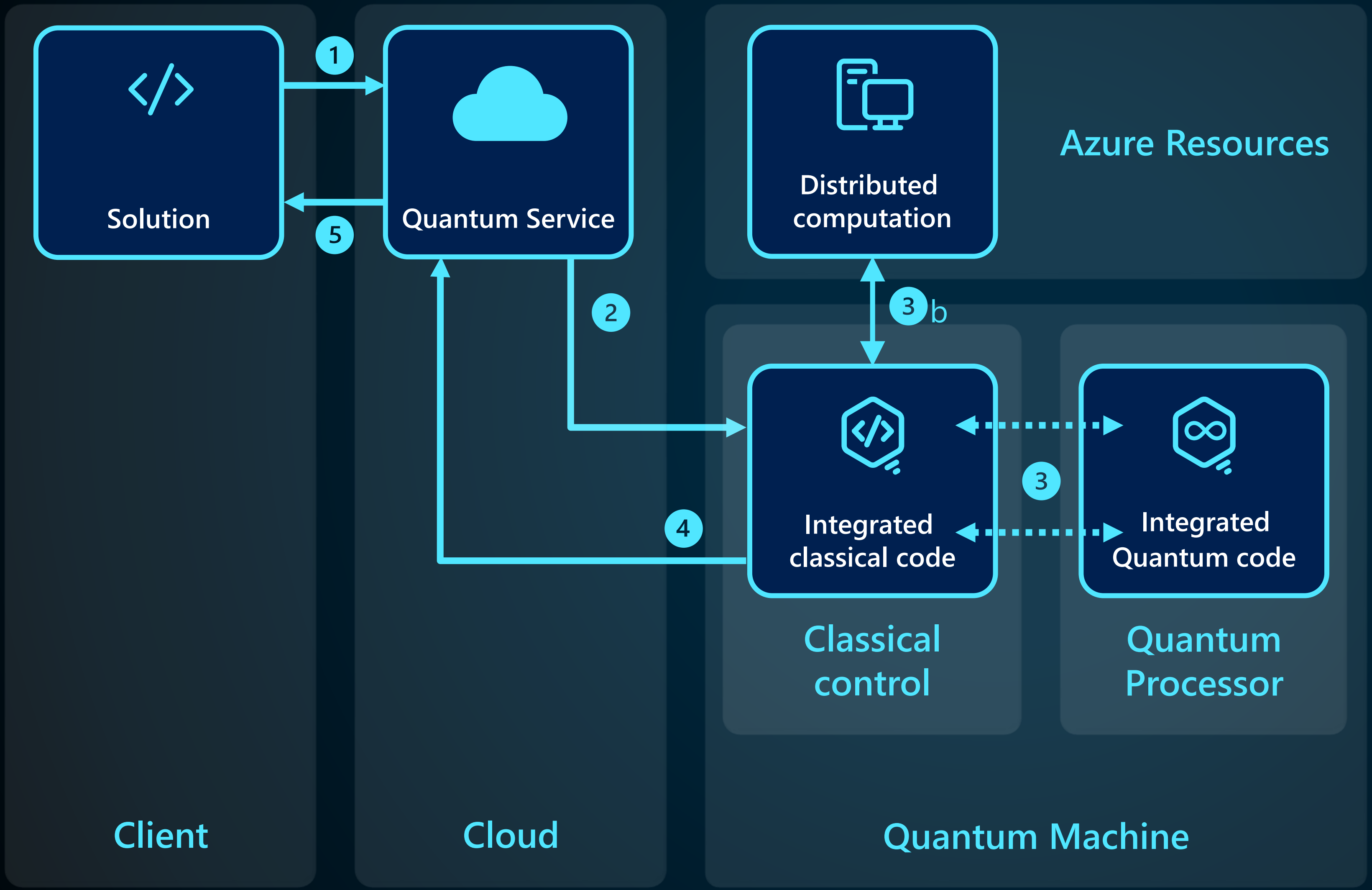
Full classical control

Examples

- Repeat until success gadgets
- Rejection sampling

Level 5: Distributed quantum computing

Distributed quantum/classical processing



3b Real-time cloud processing

Examples:

- Complex materials modelling
- Catalysis

QIR: Quantum Intermediate Representation

- Intermediate representation that expresses quantum and classical computations together.
- Frontends:
  - Qiskit
  - Q#
  - Cirq
  - …
- Backends:
  - Superconducting qubits
  - Photonic qubits
  - Trapped ions and cold atom qubits
  - Topological qubits
  - Simulators
  - Resource estimators
  - …

Q2B2021 presentation: [Q2B 2021 | Empowering Heterogeneous Quantum Computing with QIR | Panel - YouTube]

QIR: Quantum Intermediate Representation

- Intermediate representation that expresses quantum and classical computations together.
- Frontends:
  - Qiskit
  - Q#
  - Cirq
  - …
- Optimizations :
  - Of a hybrid program
  - Hardware agnostic first
  - Hardware specific then
  - Reuses LLVM tooling
- Backends:
  - Superconducting qubits
  - Photonic qubits
  - Trapped ions and cold atom qubits
  - Topological qubits
  - Simulators
  - Resource estimators
  - …

Q2B2021 presentation: Q2B 2021 | Empowering Heterogeneous Quantum Computing with QIR | Panel - YouTube

QIR: Quantum Intermediate Representation

- Intermediate representation that expresses quantum and classical computations together.

- Frontends:
  - Qiskit
  - Q#
  - Cirq
  - …

- Backends:
  - Superconducting qubits
  - Photonic qubits
  - Trapped ions and cold atom qubits
  - Topological qubits
  - Simulators
  - Resource estimators
  - …

- Optimizations :
  - Of a hybrid program
  - Hardware agnostic first
  - Hardware specific then
  - Reuses LLVM tooling

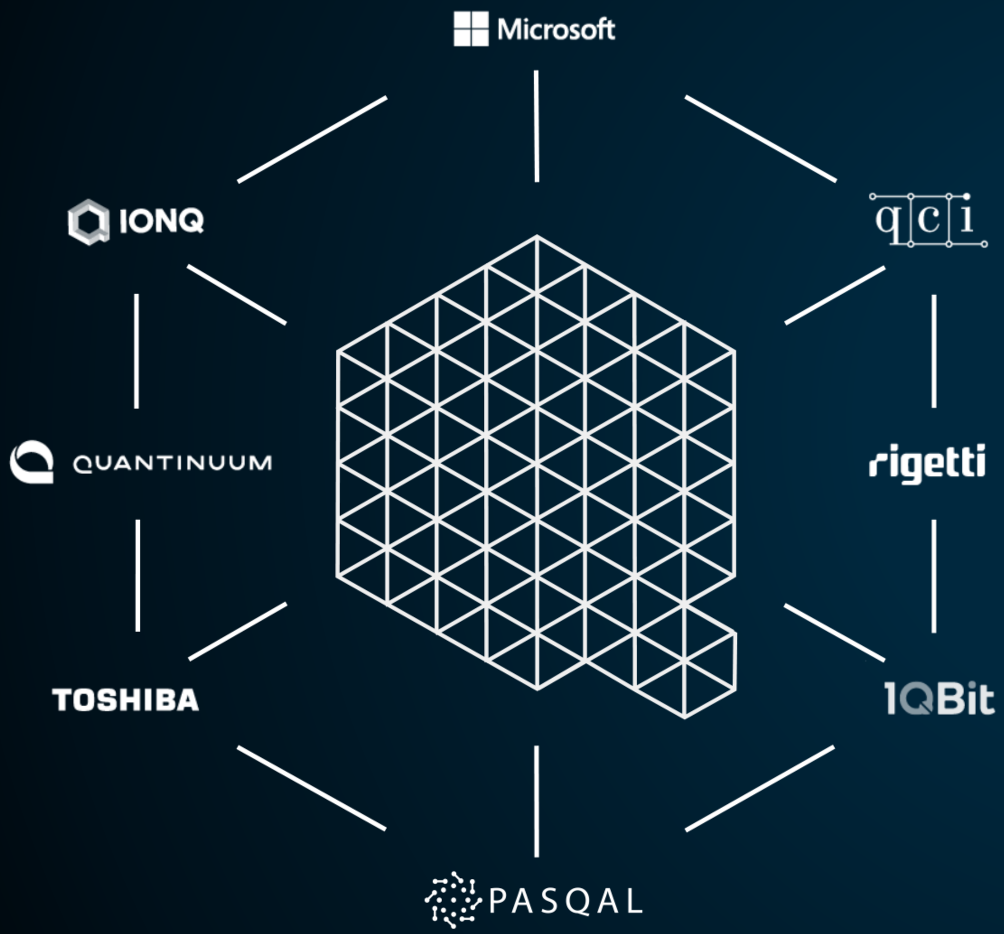
- Joint effort :
  - Quantinuum
  - QCI
  - Rigetti
  - Microsoft
  - Oak Ridge National Laboratory
  - Nvidia
  - …

Q2B2021 presentation: [Q2B 2021 | Empowering Heterogeneous Quantum Computing with QIR | Panel - YouTube]

Microsoft

Azure Quantum

The High-Performance hybrid QC cloud platform



- ✓ Unique high-performance hybrid capabilities
- ✓ Start path to fault-tolerance with Resource Estimation
- ✓ Choice of hardware from a single cloud service
- ✓ Q#, Qiskit and Cirq support
- ✓ $500 free credits for all users, larger grants up to $10,000
- ✓ Write once, run on multiple platforms

Microsoft

Thank you