

Qiskit ~~1.0~~^{1.3} Overview

Boseong Kim
Workforce and Education
IBM Quantum

Qiskit SDK

The lingua franca of quantum computing;
write once and execute quantum circuits on
10+ different hardware providers



Quantum SDK Preferred
(2024 Unitary Fund Survey)

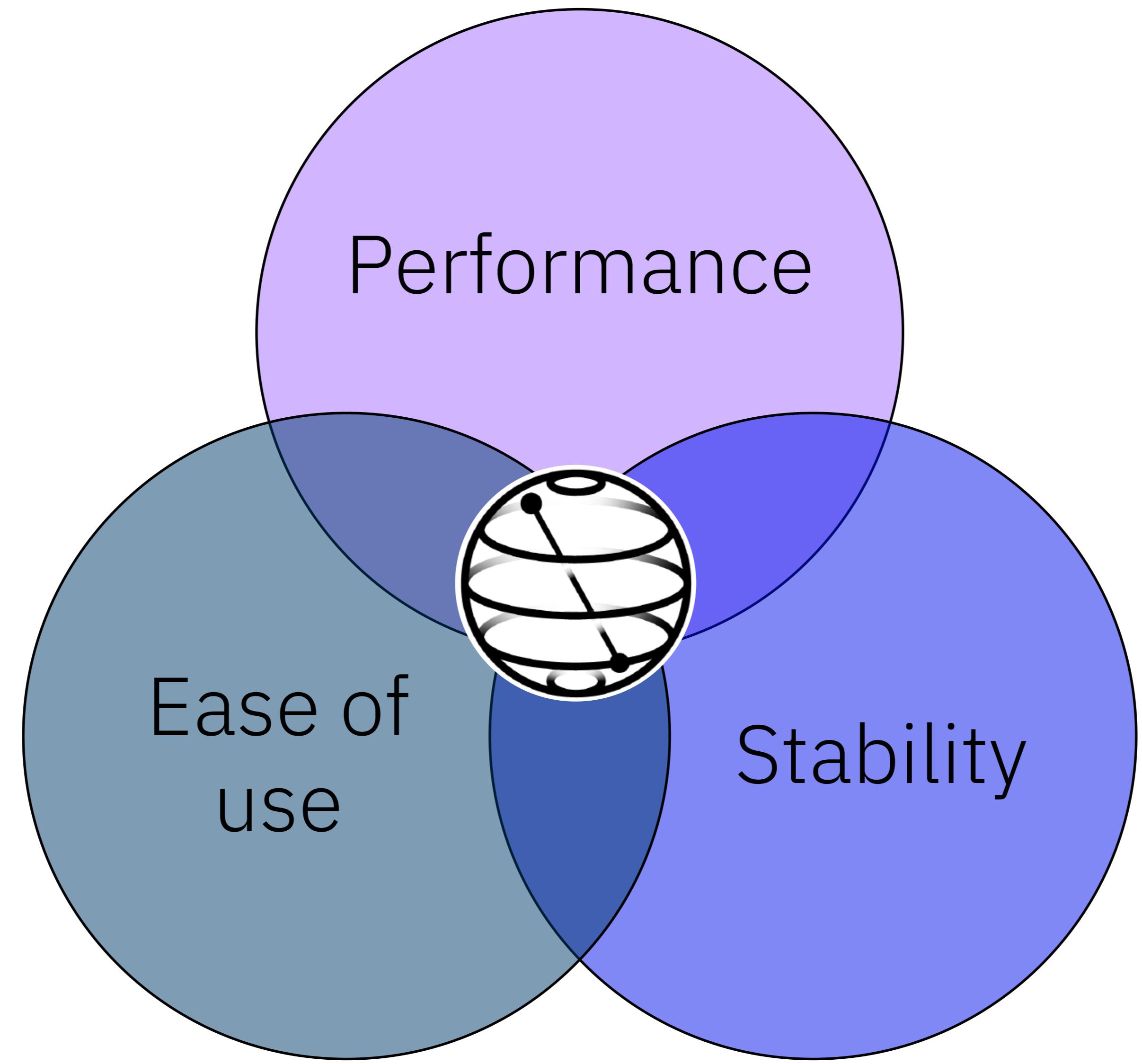
74%

Qiskit contributors that
are external to IBM

74%

Dependent Qiskit Projects

5185



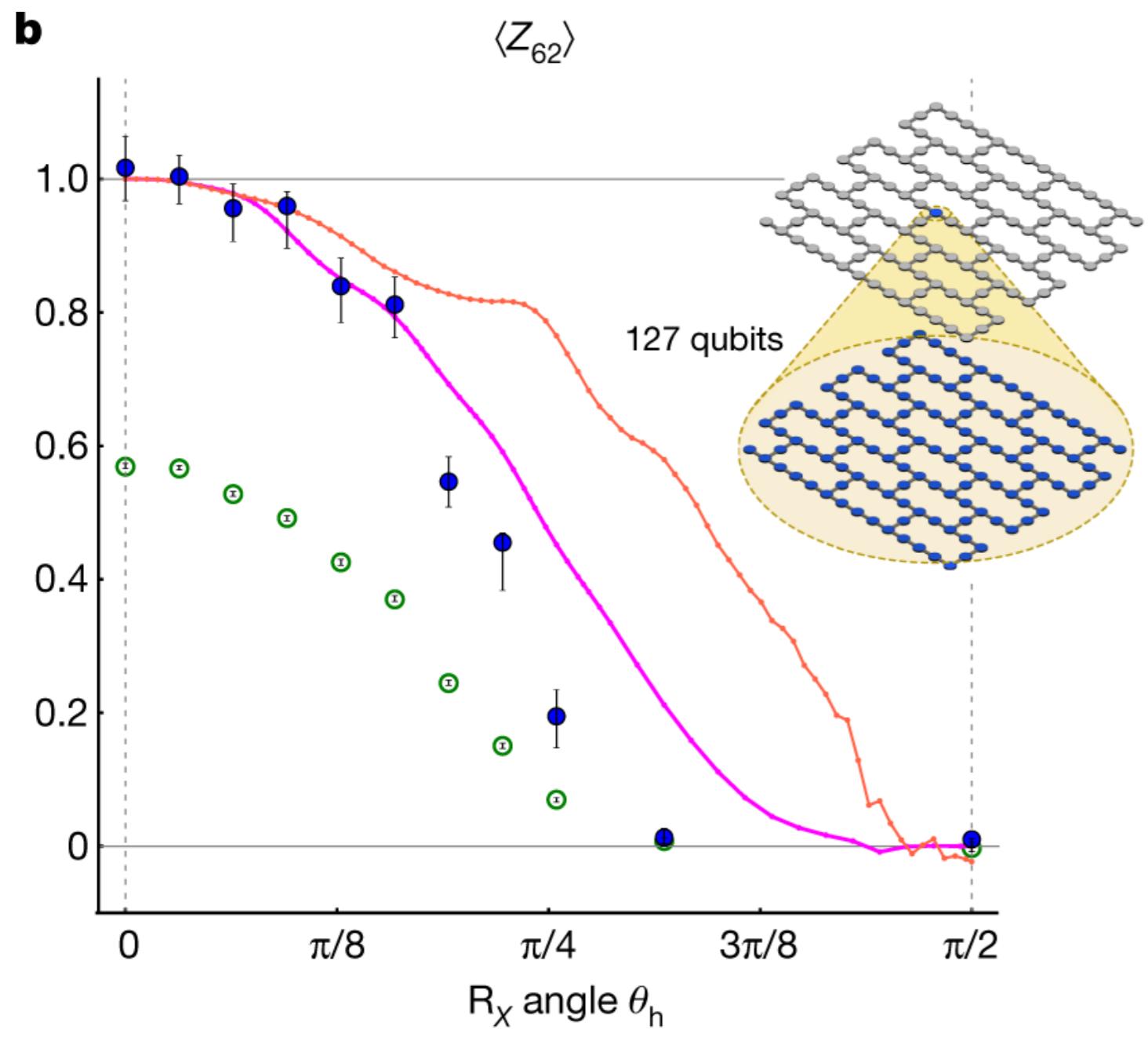
IBM Quantum

Performance

IBM

Getting to 5K Gates: Research to Product

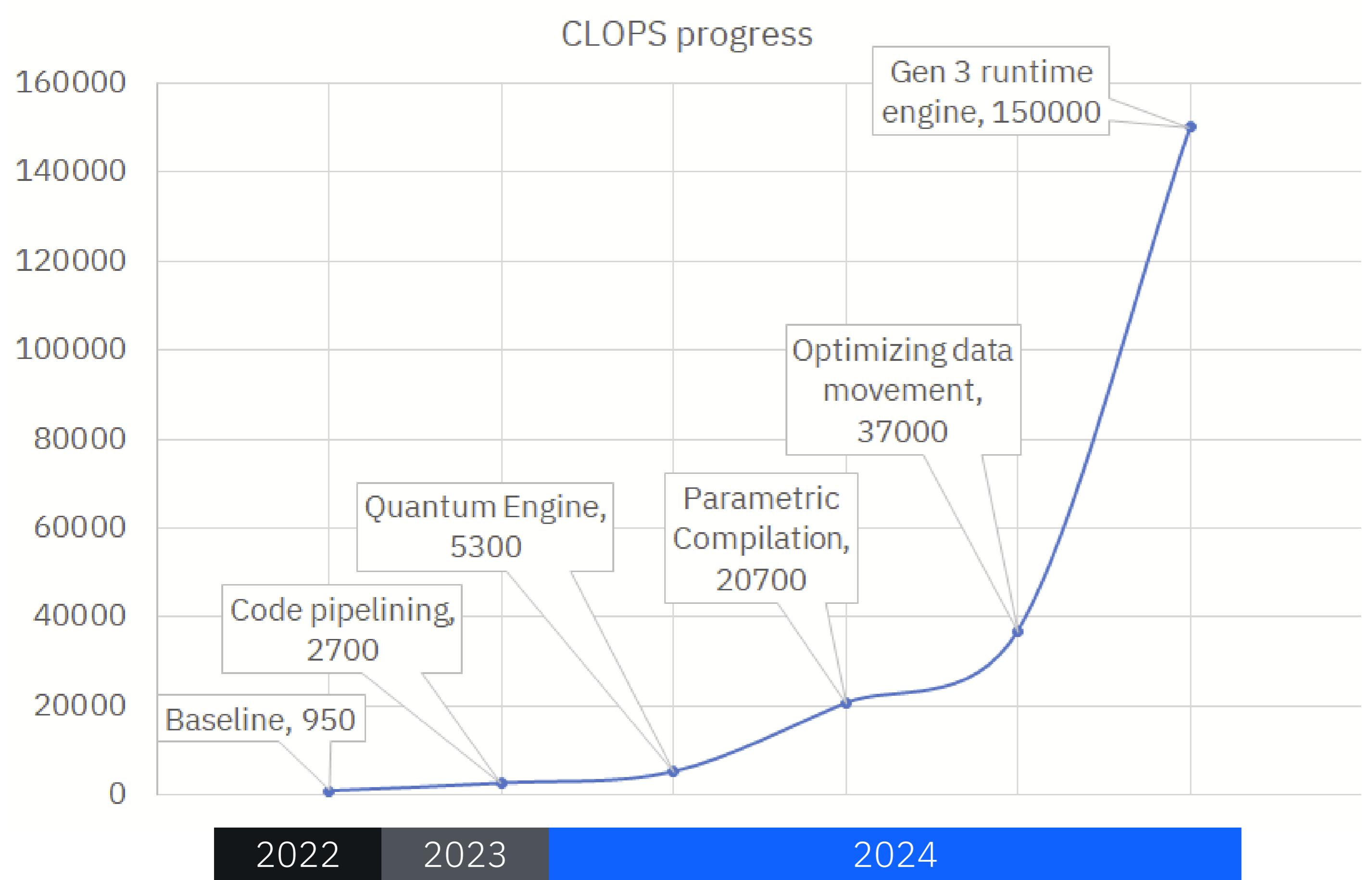
June 2023: Research demonstration of error mitigation on one observable for circuits with 2,880 2Q gates. Runtime: 122 hours



2024 5K Product:
Required progress in
quality, speed ... and
reliability

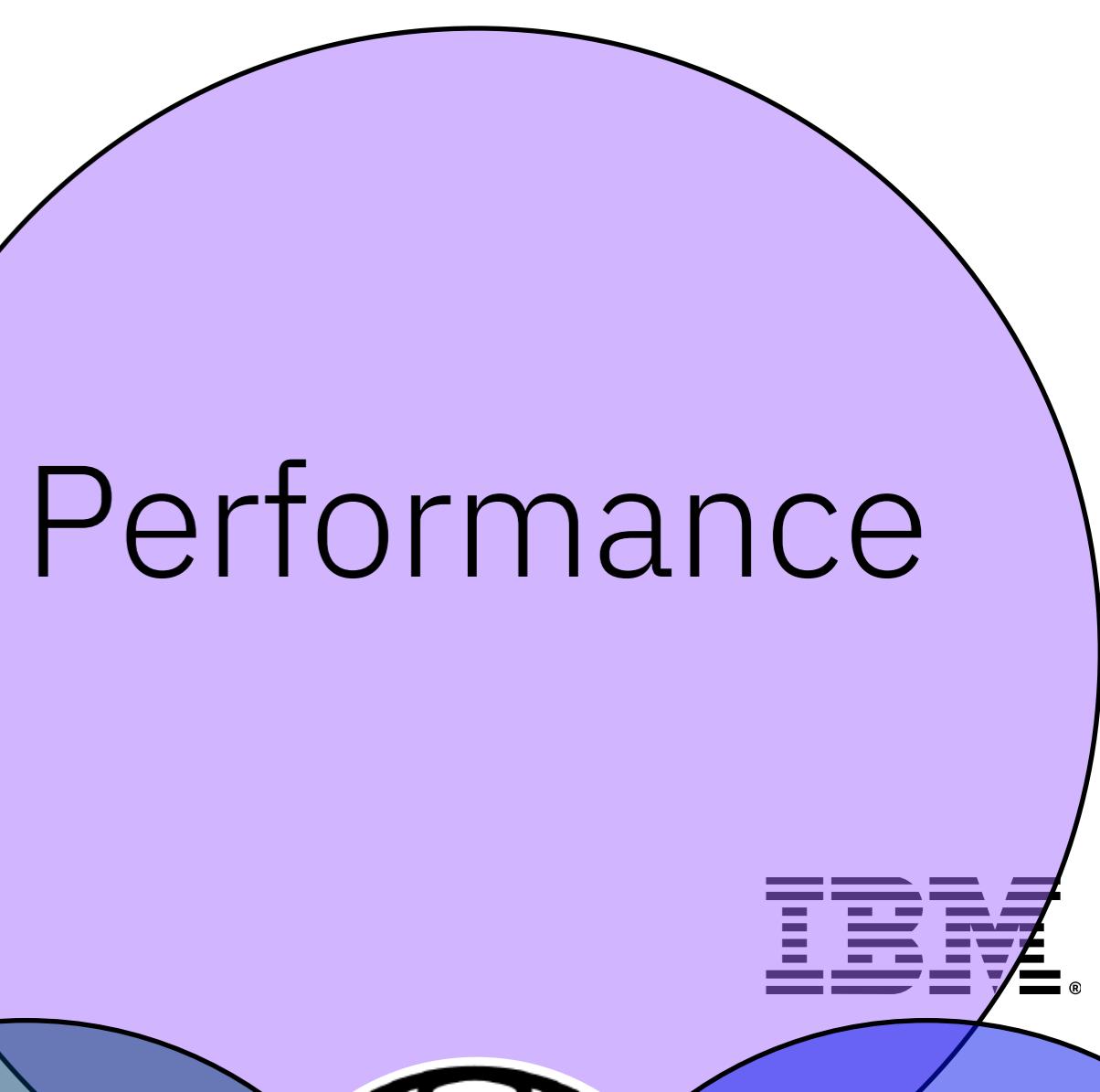
Performance

Reliable software for utility-scale exploration

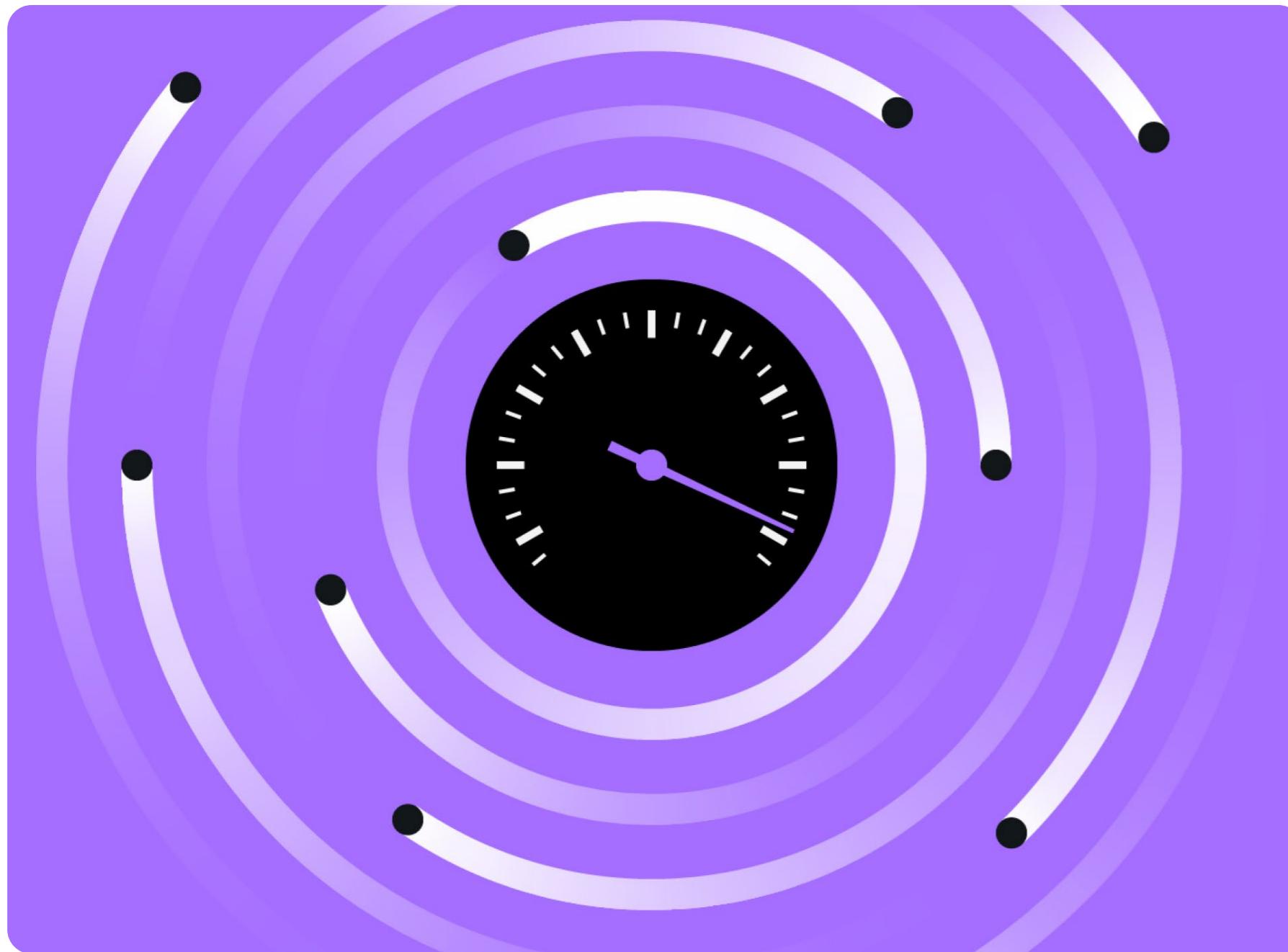


Speed up in execution of
Utility experiment

↑ 50X



Benchpress is an open and transparent benchmarking platform for quantum software that enables holistic testing of circuit creation, manipulation, and transpilation.



1066

Tests designed for Benchpress or from third-party libraries

930

Max qubits tested and $O(10^6)$ two-qubit gates

Tested Qiskit not only against itself but against all other supported quantum SDKs as well:

BQSKit

Braket

Cirq

Qiskit Transpiler Service (QTS)

Staq

Tket

Qiskit SDK

By the numbers

Qiskit 1.3 vs Tket 1.34 on transpilation tests in Benchpress:

Mean reduction in
2Q-gate count

↓ 21%

Mean transpilation
time improvement

↑ 58x

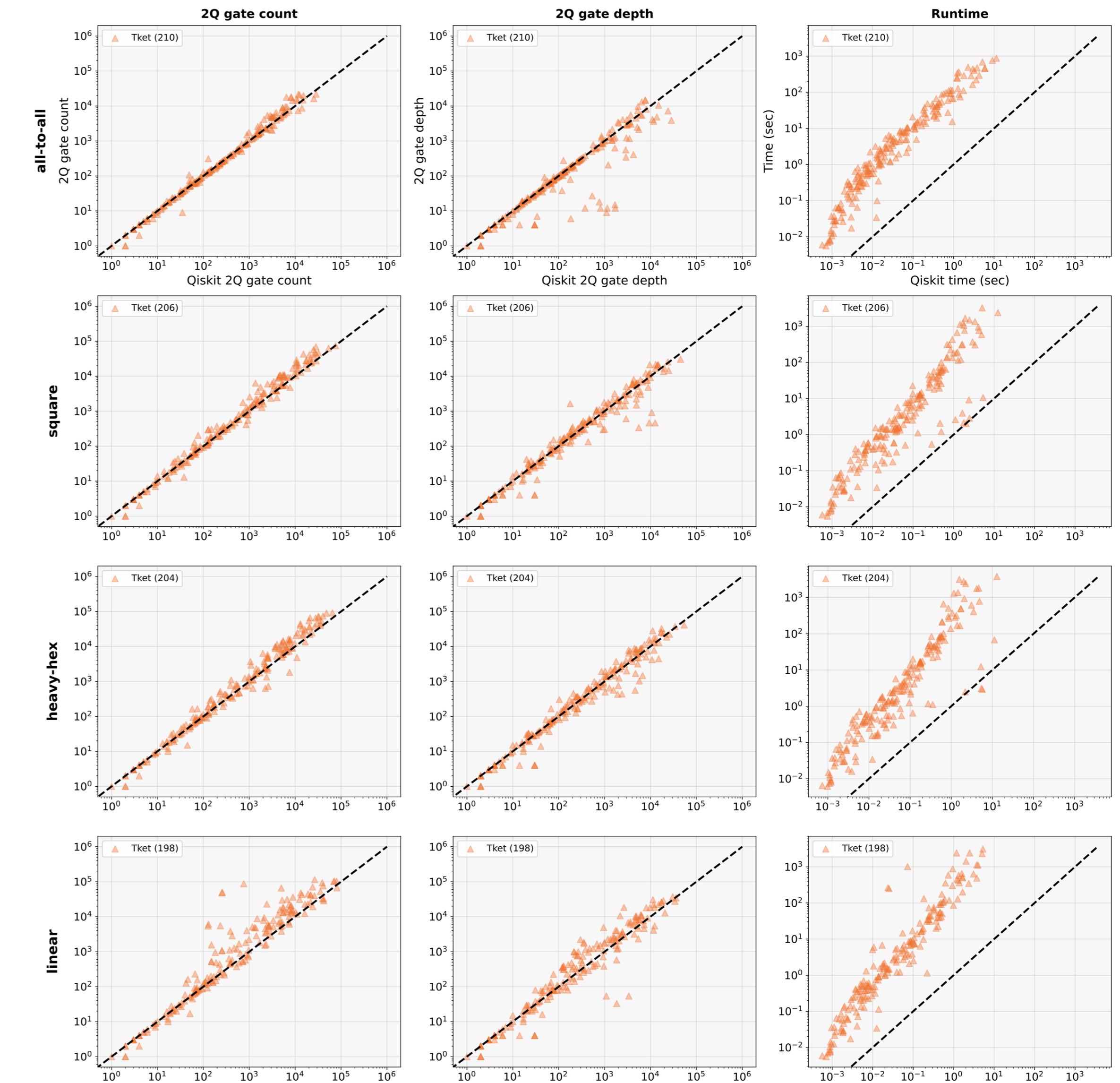
Qiskit 1.3 vs BQSKit 1.1.2 on transpilation tests in Benchpress:

Mean reduction in
2Q-gate count

↓ 18%

Mean transpilation
time improvement

↑ 514x



Qiskit Transpiler Service and AI Passes

With the Qiskit Transpiler Service, you can transpile on the cloud and leverage the power of the AI transpiler passes.

This time we bring three major updates

↓

01

Improved stability & performance

Tested on 1000 qubits and 1M gates, 30%+ depth improvement on benchpress circuits.

02

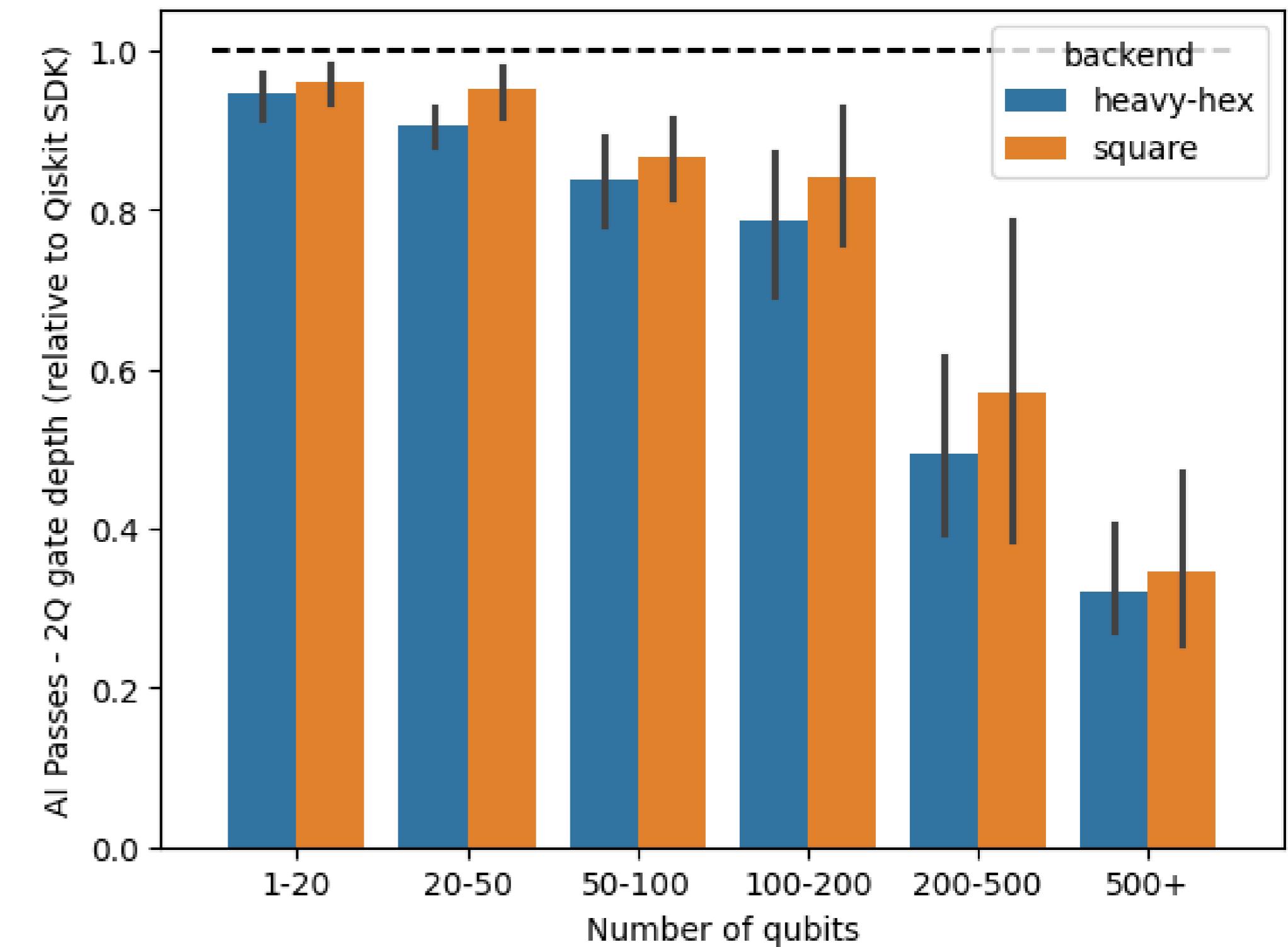
New AI Synthesis Pass

Pauli Network synthesis, perfect for chemistry circuits.

03

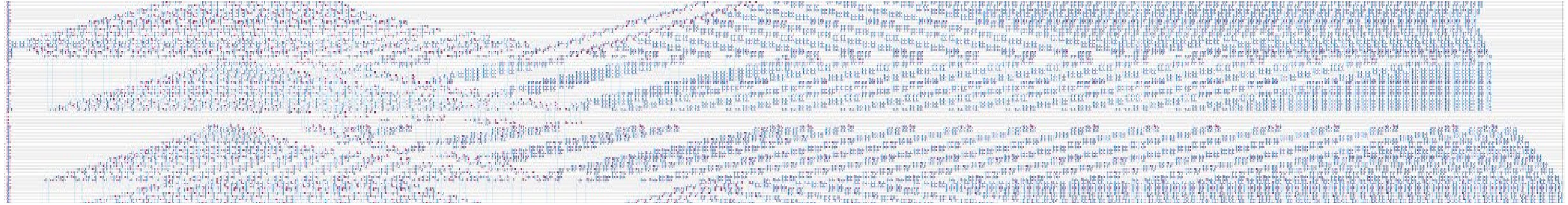
Local AI mode

You can now use some of the AI passes locally!



Circuit depth for circuits transpiled with Qiskit+AI, relative to standard Qiskit, for the Benchpress circuits (lower is better)

Sample-based quantum diagonalization (SQD) with 5K gates: Hydrogen Chains (H₂n)



Circuit Stats:

ibm_fez

Qubits: 84

2Q gates: ≈ 5100

1Q Gates: ≈ 16600

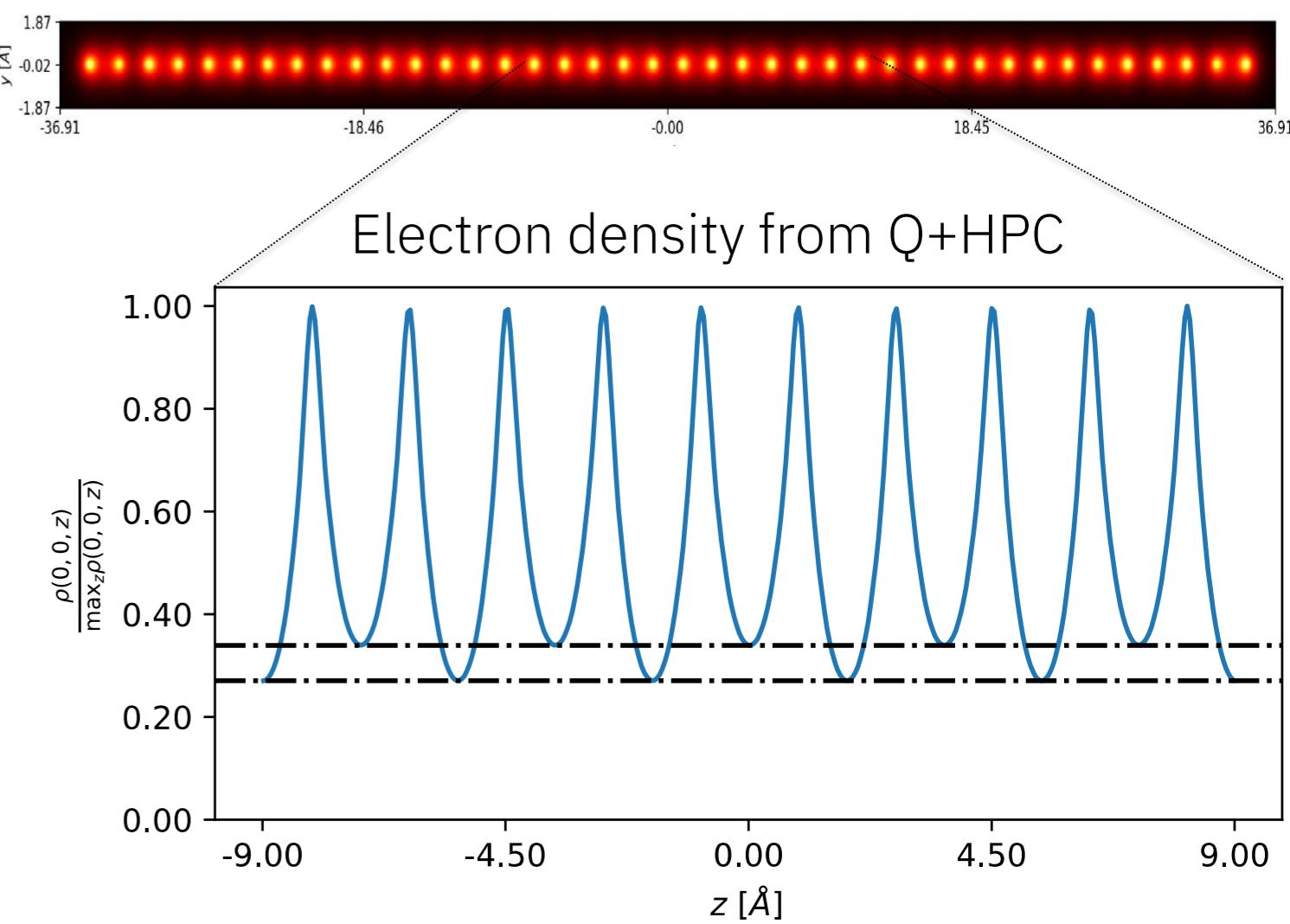
Duration: $\approx 28 \mu s$

IBM Quantum

Improving outcomes



Extracting impactful properties



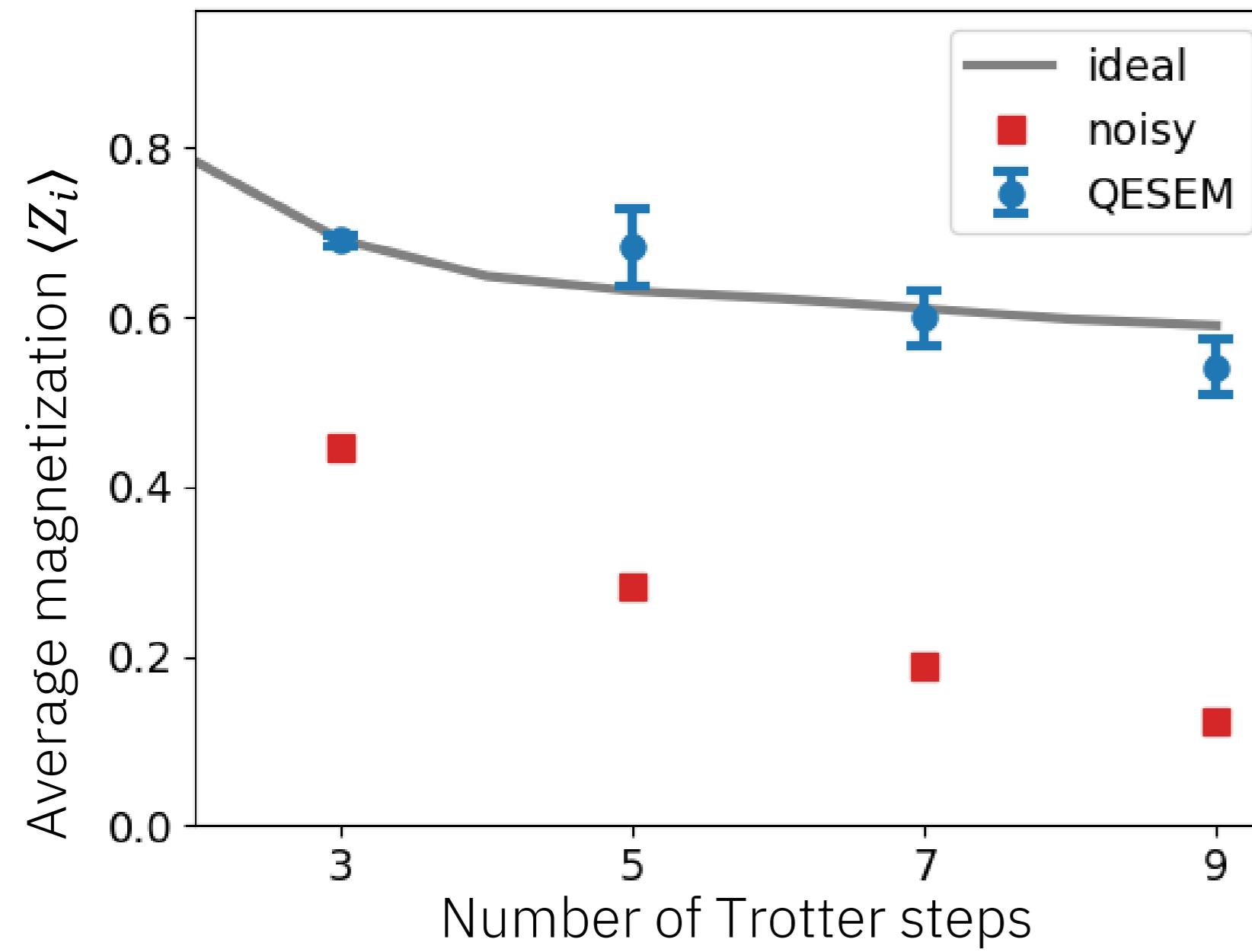
Reaching 5K scale with Partners



Qedma

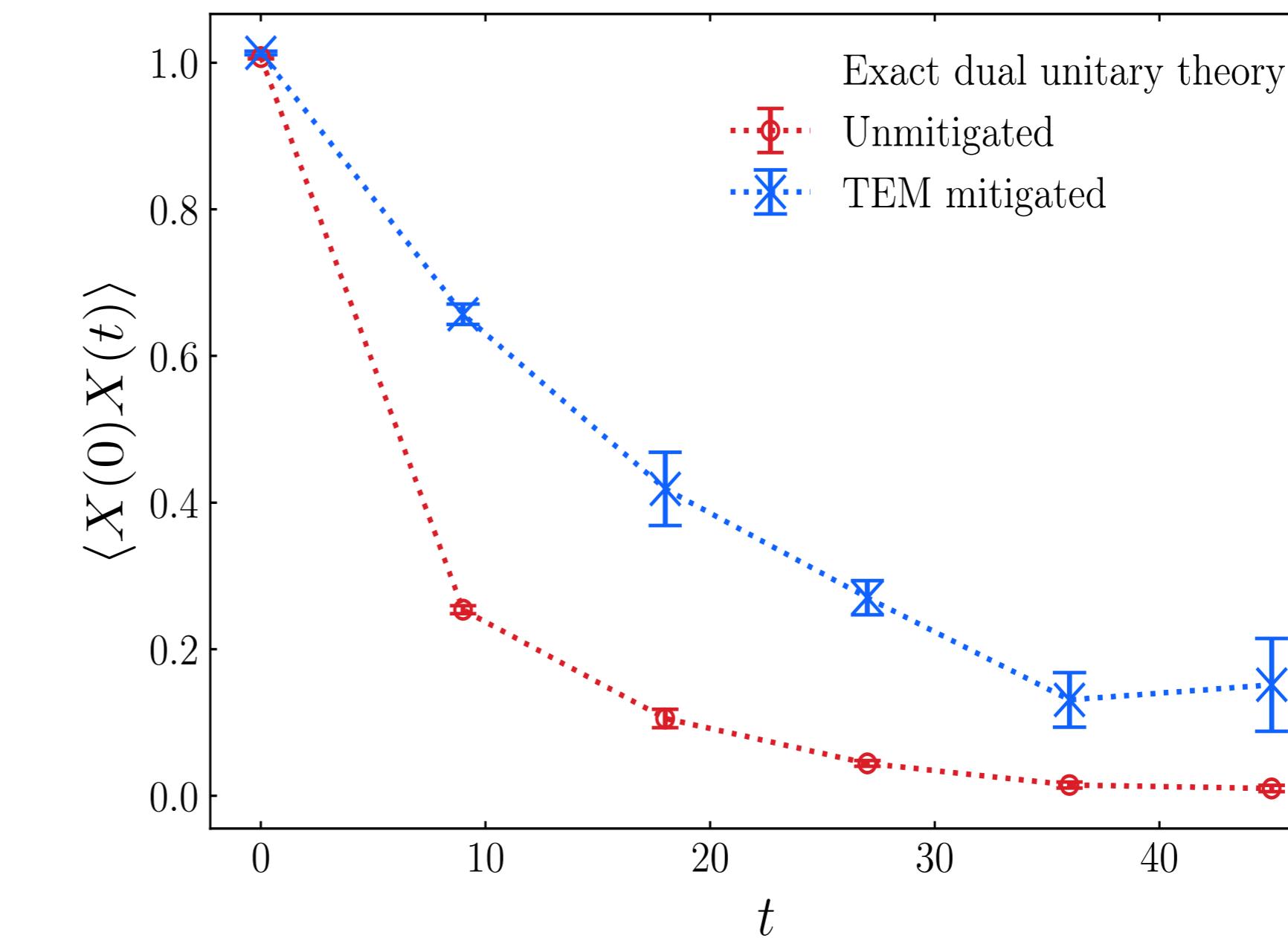
QESEM method produces *unbiased* expectation values

119-qubit 2D Ising model
up to 2.3K gates



TEM method performs error mitigation in classical post-processing

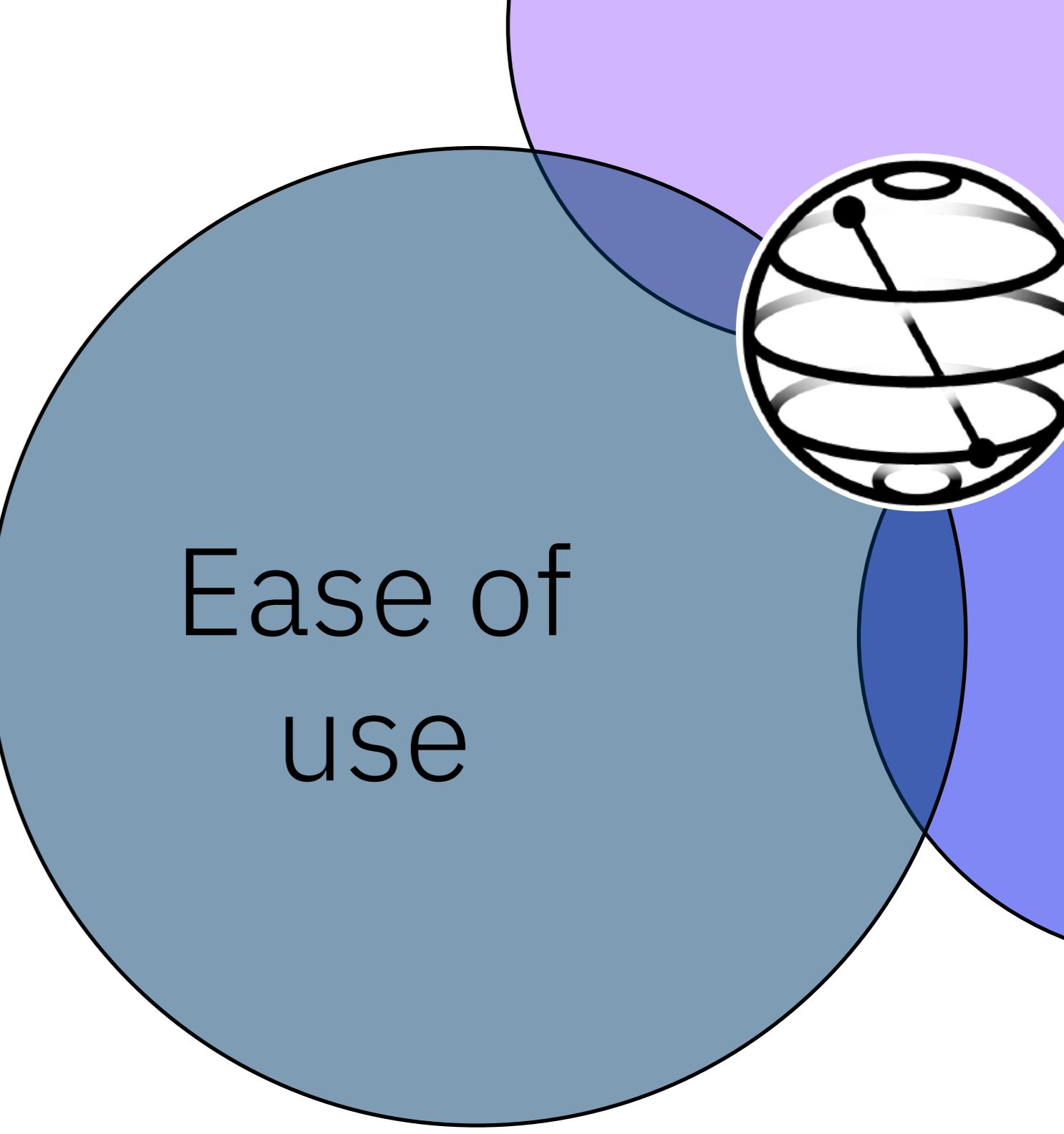
91-qubit dual unitary circuit
with up to 4.1K gates



IBM Quantum

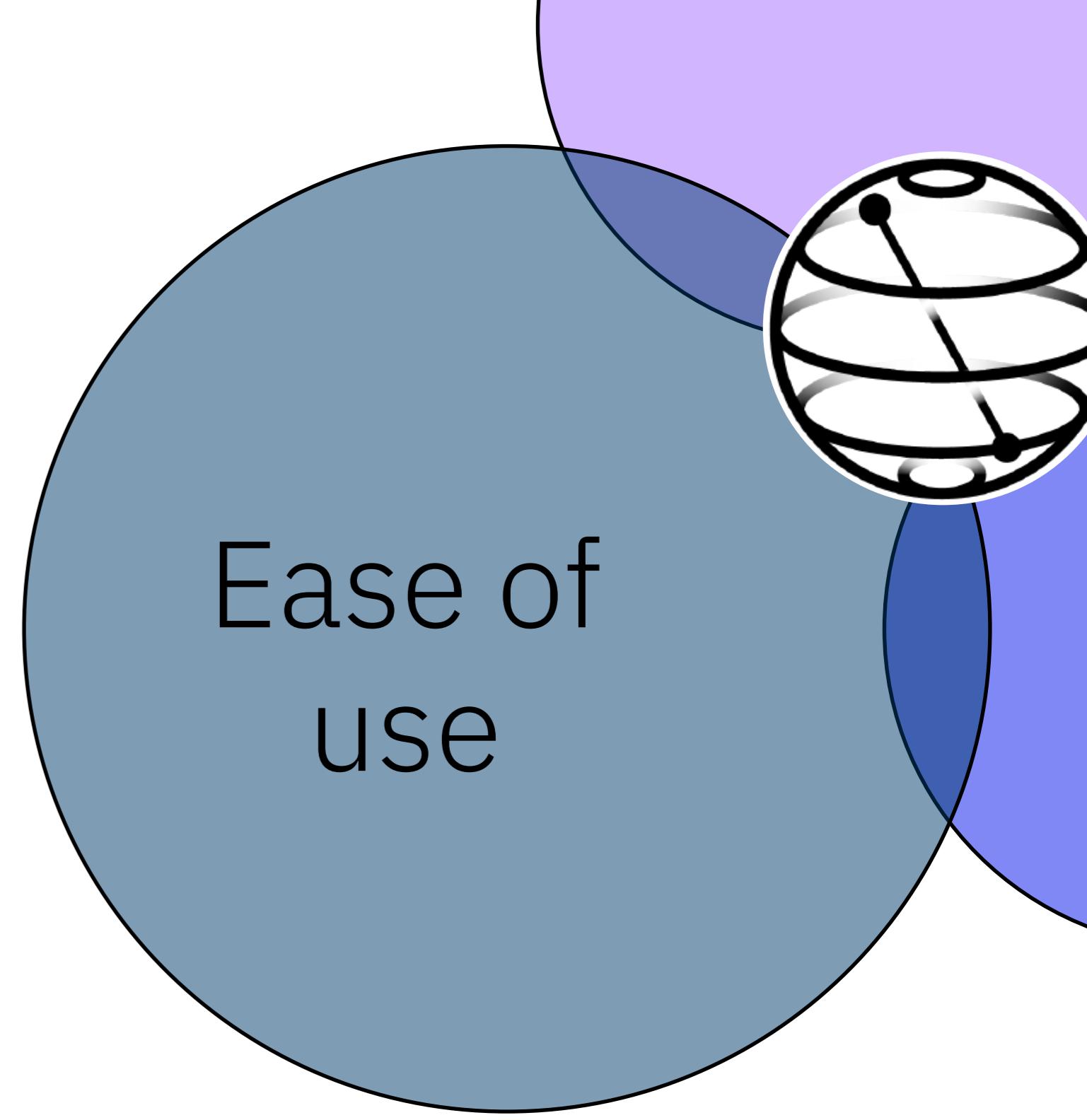
Performance

IBM



Ease of
use

Qiskit Pattern: The anatomy of a quantum algorithm



Ease of
use

01

Map problem instance
to quantum circuits and
operators

Q^+

Map

02

Optimize for target
hardware execution

\vec{x}

Optimize

03

Execute via
Qiskit Runtime

\mathbb{E}

Execute

04

Result processing

\swarrow

Post-Process

Qiskit SDK sets the foundation

Qiskit SDK gives us a base layer of building blocks for building and running quantum algorithms

Qiskit Circuit Library

Input:
Domain inputs

Output:
Circuits, observable

Q^+
Map

Transpiler

Input:
Circuits, observable

Output:
ISA circuit, observable

\vec{x}
Optimize

Primitives

Input:
ISA circuit, observable

Output:
Expectation value/samples

\mathbb{E}
Execute

Quantum Info

Input:
Expectation value/samples

Output:
Data objects/visualizations

\curvearrowleft
Post-Process

Qiskit addons build on the Qiskit SDK

A collection of research capabilities developed as modular tools that can plug into a workflow to design new algorithms at the utility scale

Starting with multi-product formulas ([MPF](#)), approximate quantum compilation ([AQC-Tensor](#)), operator backpropagation ([OBP](#)), and sample-based quantum diagonalization ([SQD](#)).

AQC-Tensor

MPF

Qiskit Circuit Library

OBP

Circuit cutting

Transpiler

SQD

M3

Quantum Info

Input:
Domain inputs

Output:
Circuits, observable

Q^+
Map

Input:
Circuits, observable

Output:
ISA circuit, observable

\vec{x}
Optimize

Input:
ISA circuit, observable

Output:
Expectation value/samples

\square
Execute

Input:
Expectation value/samples

Output:
Data objects/visualizations

\swarrow
Post-Process

Accelerating discovery with Qiskit addons

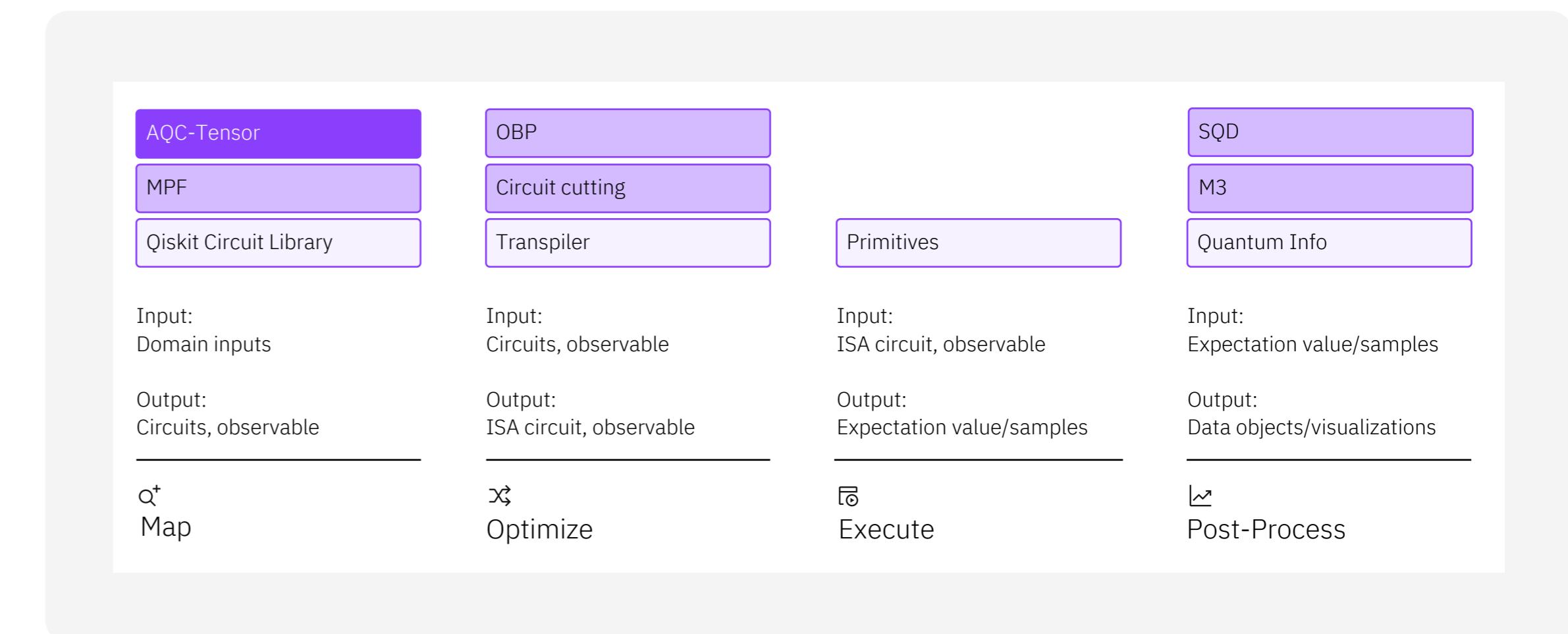
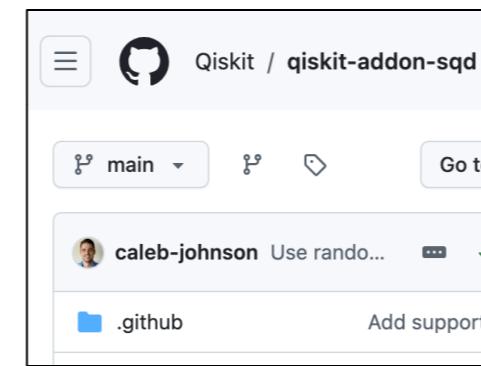
Qiskit addons - a collection of research capabilities developed as modular tools that can plug into a workflow to design new algorithms at the utility scale, starting with:

Multi-product formulas (MPF)

Approximate quantum compilation (AQC-Tensor)

Operator backpropagation (OBP)

Sample-based quantum diagonalization (SQD)



Faster research
Up to:

↑ 5X

1 year of research

“Chemistry Beyond Exact
Solutions on a Quantum-
Centric Supercomputer”

arXiv:2405:05068



2.5 months of research

“Accurate quantum-centric
simulations of supramolecular
interactions”

arXiv:2410:09209



“Quantum-Centric Study of
Methylene Singlet and Triplet
States”

arxiv.org/abs/2411.04827



Introducing: Qiskit Functions Catalog

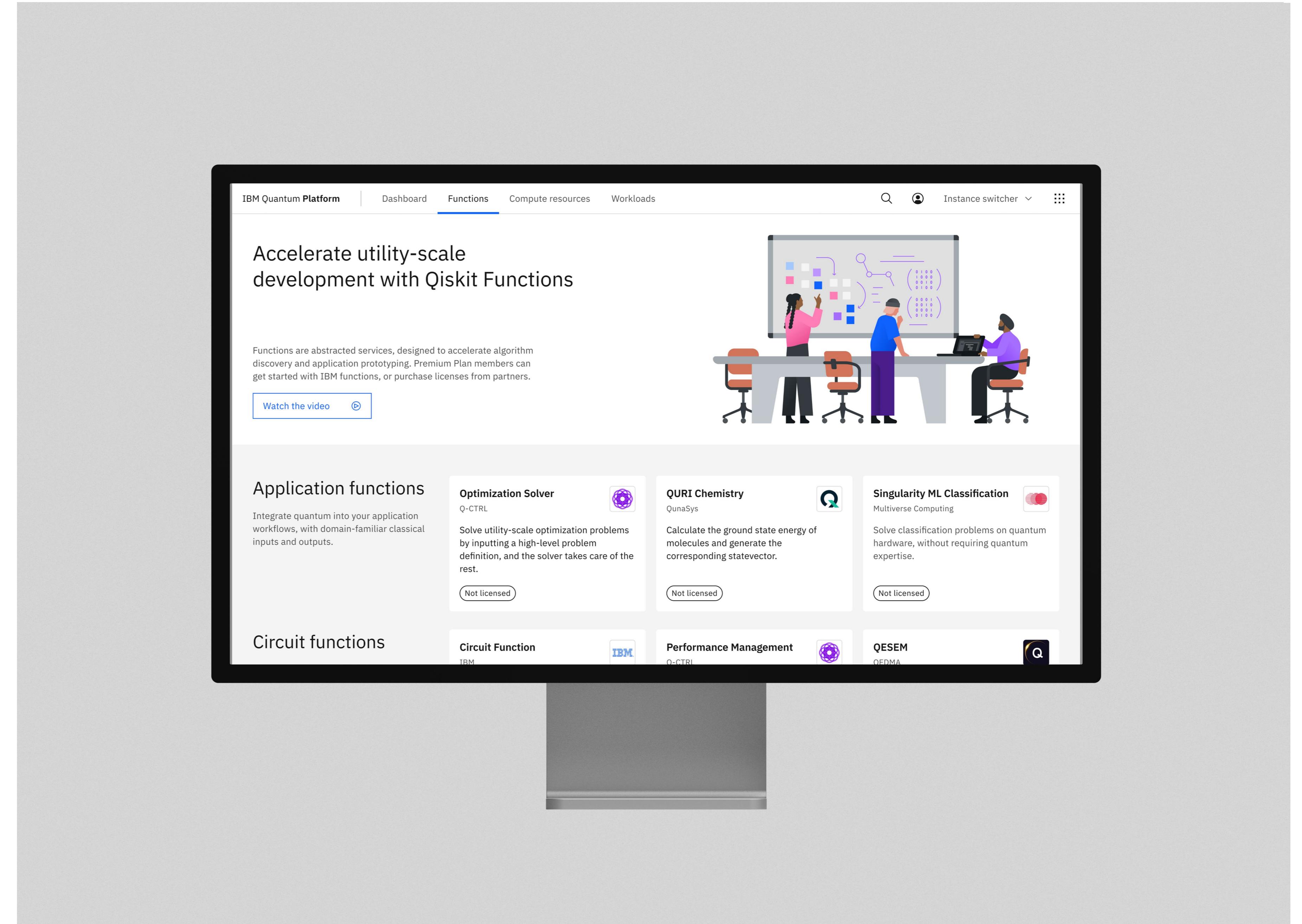
A catalog of abstracted services to accelerate utility-scale research, by IBM and third-party partners.

3 application functions

- Chemistry
- Optimization
- Machine Learning

4 circuit functions

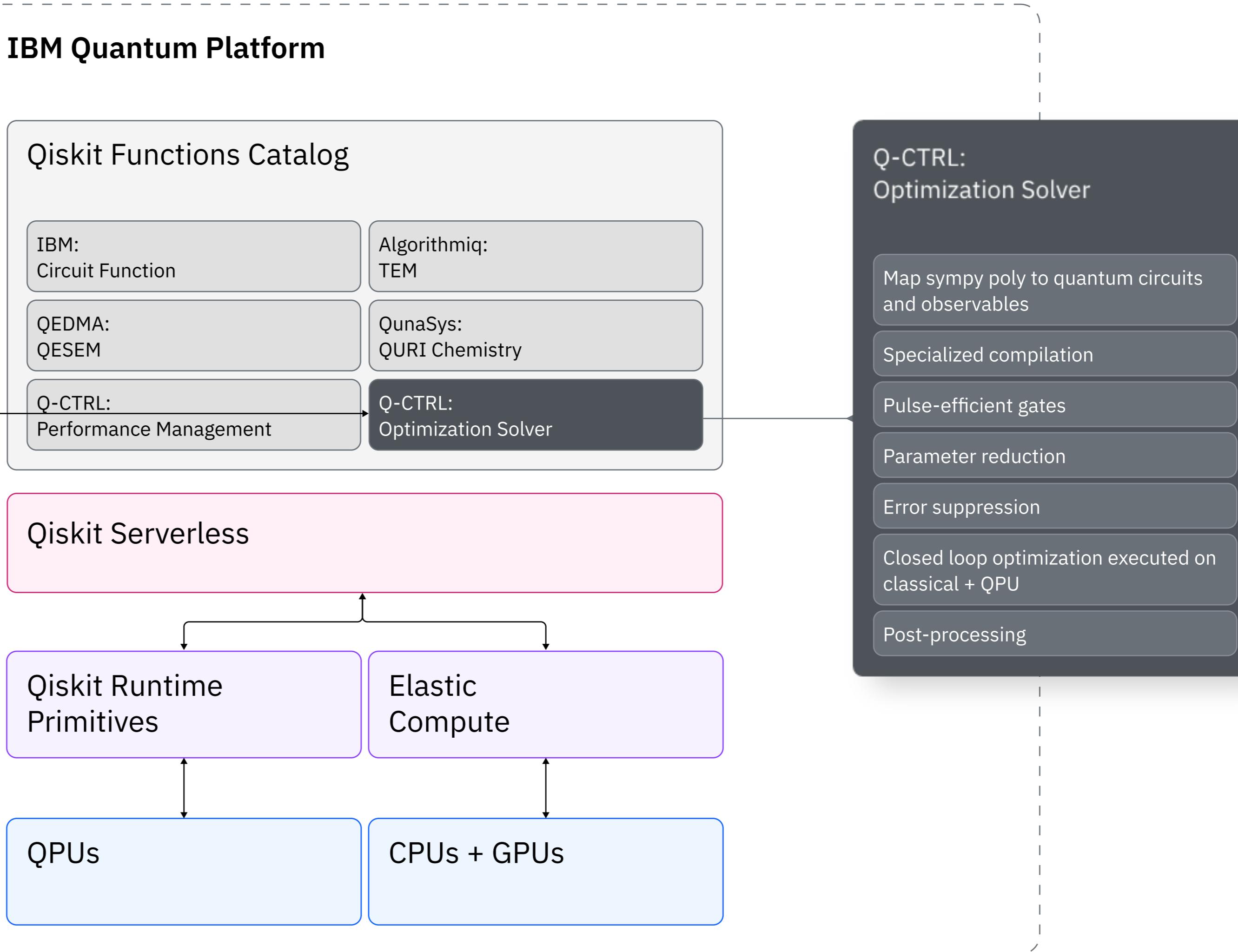
- Base functions to accelerate research towards future application functions



Exploring the Qiskit Functions Catalog

Qiskit IBM Catalog
+ Sympy

```
00 x = symbols([f"x[{i}]" ...)  
    poly_func = Poly(-4*x[0]*x[1] ...)  
  
    Poly(-4x[0]x[1]-8x[1]x[2]x[3] ...)  
  
01 catalog = QiskitFunctionsCatalog(  
    token=token  
)  
  
02 solver = catalog.load(  
    'q-ctrl/optimization-solver'  
)  
  
03 job = solver.run(  
    problem=srepr(poly_func),  
    instance=instance  
)  
  
04 result = job.result()  
result["solution_bitstring_cost"]  
  
-242.0
```



Qiskit Code Assistant

Enabling developers to write quantum computing code frictionless and faster with GenAI. Powered by our granite-8b-qiskit model and watsonx.

Highlights:

- Service available to our IBM Quantum Premium plan users
 - Open-source model granite-8b-qiskit publicly available
 - IDE extensions available for JupyterLab and Visual Studio Code
 - Evaluated using our open source Qiskit HumanEval benchmark



arXiv: 2406.14712

The screenshot shows a code editor window with the following details:

- Title Bar:** qiskit-code-assistant.py U
- Status Bar:** main* ↻ ⌂ ✘ 0 ⚠ 0 Qiskit Code Assistant: granite-8b-qiskit-rc-0.5 UTF-8 LF {} Python 3.11.3 ('env': venv) 🔔
- Sidebar (Left):** A vertical toolbar with icons for file operations (New, Open, Save, etc.), search, navigation, and other development tools. Some icons have numerical counts (e.g., 1, 3).
- Code Editor Area:** The main pane displays the following Python code:

```
1 # Let's build a variational quantum eigensolver (VQE) using Qiskit.
```
- Header Bar (Top Right):** Includes icons for navigation, search, and other application functions.

Abstraction
Services

Qiskit Functions Catalog

Circuit Functions Application Functions

Dev Tools

Heterogeneous
Orchestration Layer

Compute

Qiskit SDK

Qiskit addons

Qiskit Transpiler
Service

Qiskit Serverless

Qiskit ↔ HPC
Integration

Quantum Computer

Qiskit Runtime

Quantum Processing Unit

Quantum Chip

High Performance
Classical Compute

CPU

GPU

AIU

Qiskit
Code
Assistant

Abstraction
Services

Dev Tools

Heterogeneous
Orchestration Layer

Compute

IBM Quantum Platform

Qiskit Functions Catalog

Circuit Functions Application Functions

Qiskit SDK

Qiskit addons

Qiskit Transpiler
Service

Qiskit Serverless

Qiskit ↔ HPC
Integration

Quantum Computer

Qiskit Runtime

Quantum Processing Unit

Quantum Chip

High Performance
Classical Compute

CPU

GPU

AIU

Qiskit
Code
Assistant

The screenshot shows the IBM Quantum Platform dashboard. At the top, there's a navigation bar with 'IBM Quantum Platform' (highlighted), 'Dashboard', 'Functions', 'Compute resources', and 'Workloads'. On the right of the bar are a search icon, a user profile icon, the URL 'ibm-q/premium/main', and a three-dot menu icon.

The main area has a blue header with the text 'Rachel Landow' and 'IBM Quantum Platform'. Below the header is a purple banner with the text 'API token' and a redacted token value followed by copy/paste icons.

On the left, there's a 'Premium Plan' section with a button 'Explore plans'. Next to it is an 'Instance usage summary' card showing 'Your usage 6h 45m', 'Total used 7h 25m', and 'Remaining 2h 35m'. A progress bar indicates usage.

The central part of the dashboard features a 'Recent workloads' section. It shows 1 pending workload and 40 finished workloads. A table lists the details of the completed workloads:

| ID | Status | Completed | Mode | Compute resource | Usage |
|----------------------|-------------|------------------------|---------|------------------|-------------|
| csyOvbn8ex1g0csy0... | In progress | Est. wait 24 mins | Job | ibm_sherbrooke | Est. 3m 16s |
| csmr4e99vhgg0csmr... | Completed | Month DD, 20YY 1:30 PM | Job | ibm_brussels | 14m 5s |
| csmqwgicrs0g0csmq... | Completed | Month DD, 20YY 1:30 PM | Session | ibm_kawasaki | 5m 12s |
| csmqwgicrs0g0csmq... | Completed | Month DD, 20YY 1:30 PM | Batch | ibm_torino | 18m 43s |
| csmqwgicrs0g0csmq... | Completed | Month DD, 20YY 1:30 PM | Batch | ibm_kyiv | 18m 43s |

Below the table are three cards: 'Instance QPUs' (8), 'Documentation' (with a search bar and 'Hello World' example), and 'Learning' (with a course titled 'Quantum Computing in Practice').

On the right side, there's a 'What's new' section with four items:

- Service alert: 'UPDATED - Maintenance in November' (3 hours ago) • [Read more](#)
- Product update: 'Fractional gates to become part of the Heron instruction set architecture' (7 Nov 2024) • [Read more](#)
- Product update: 'What's new in the docs?' (1 Nov 2024) • [Read more](#)
- Product update: 'A closer look at Qiskit Code Assistant' (22 Oct 2024) • [Read more](#)
- Product update: 'Qiskit is the most performant quantum SDK' (15 Oct 2024) • [Read more](#)

IBM Quantum Learning

Explore new Learning Paths to build your quantum computing foundations, and run your own utility-scale experiments on IBM Quantum systems.

Courses, tutorials, and educational resources by leading quantum experts, including:



John Watrous
Technical Director
IBM Quantum Education

IBM Quantum

The screenshot shows a learning path titled "Scaling quantum computing toward utility" on the IBM Quantum Learning platform. The path consists of three sequential components: "Quantum Computing in Practice" (Course, 3 hrs), "Qiskit Addons" (Guide, 2 hrs), and "Solve Utility-Scale Quantum Optimization Problems" (Tutorial, 1 hr 30 min). The estimated completion time for the entire path is 6 h 30 min. The page includes navigation links for Home, Catalog, Network, and Composer, as well as a search bar and a sidebar with "On this page" and "Was this page helpful?" sections.

Estimated completion time: 6 h 30 min

Learn best practices of how to leverage the potential of quantum computers with 100 or more qubits to solve real-world problems.

Learning path content

```
graph LR; A[Quantum Computing in Practice] --> B[Qiskit Addons]; B --> C[Solve Utility-Scale Quantum Optimization Problems]
```

Quantum Computing in Practice
Course
3 hrs

Qiskit Addons
Guide
2 hrs

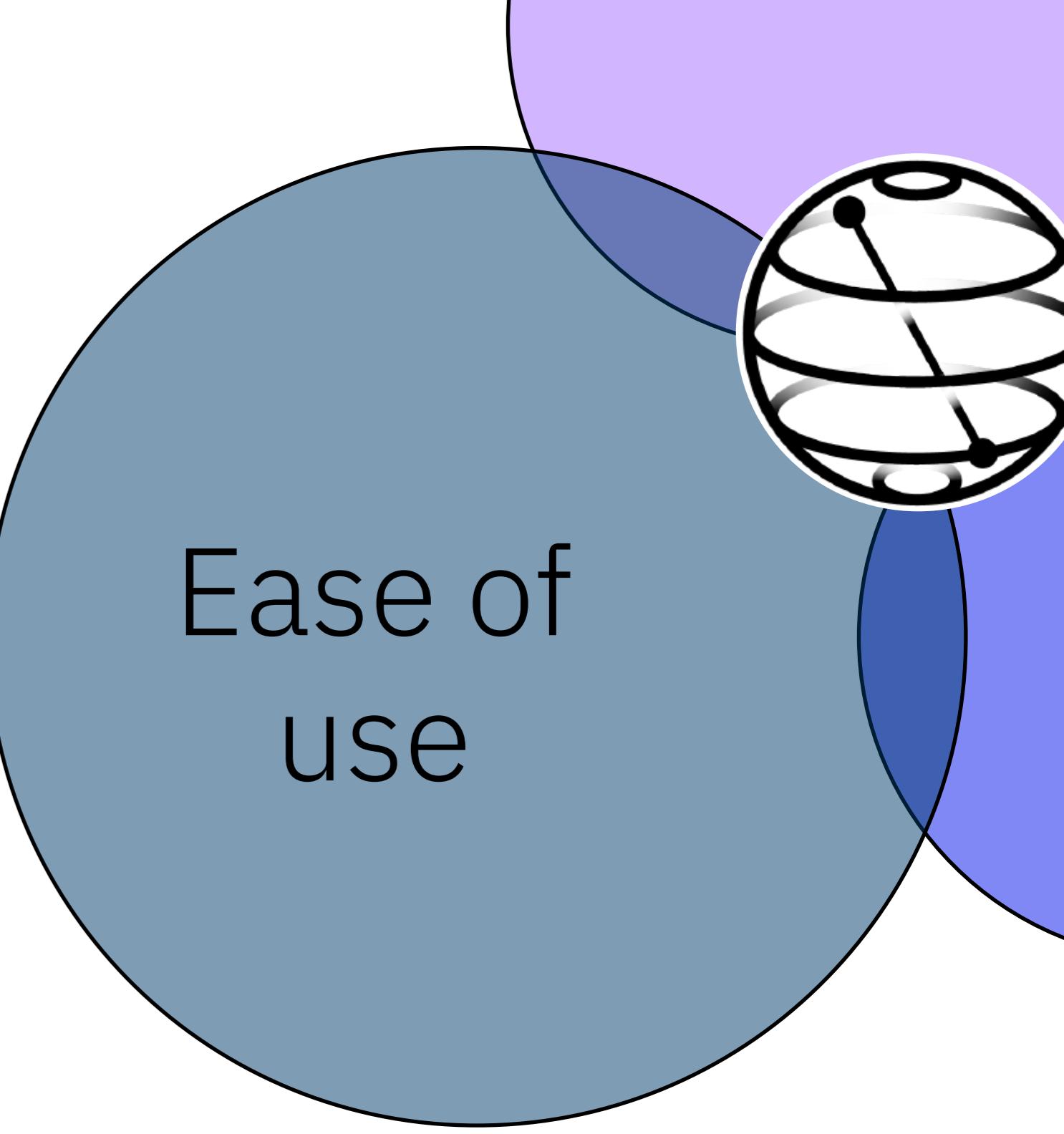
Solve Utility-Scale Quantum Optimization Problems
Tutorial
1 hr 30 min

On this page

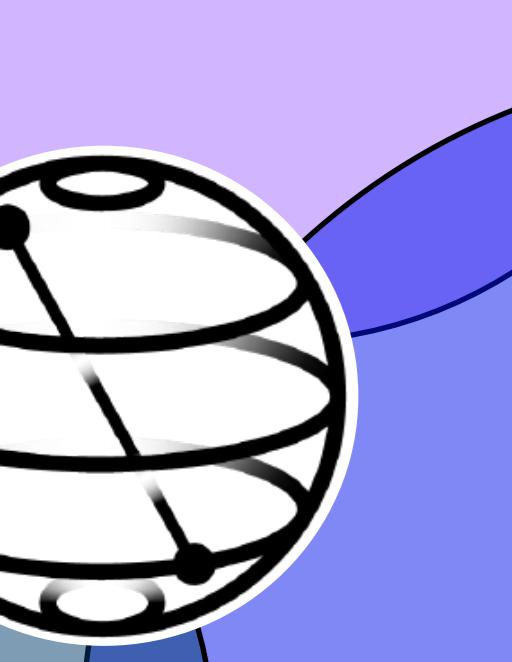
- Learning path content
- Who is this path intended for?
- Prerequisites
- To learn more
- Post-completion survey
- Next steps

Was this page helpful?

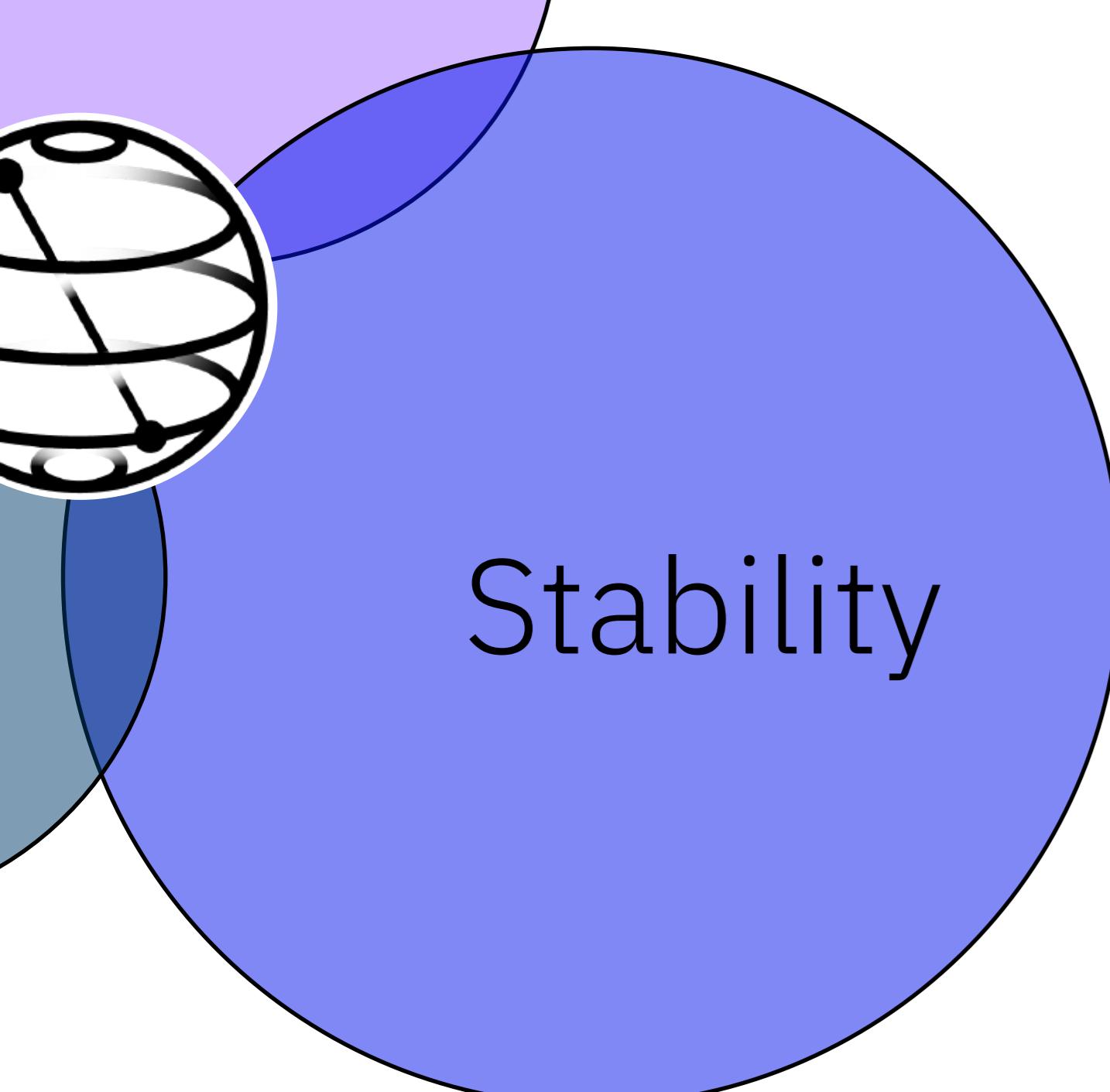
Yes No



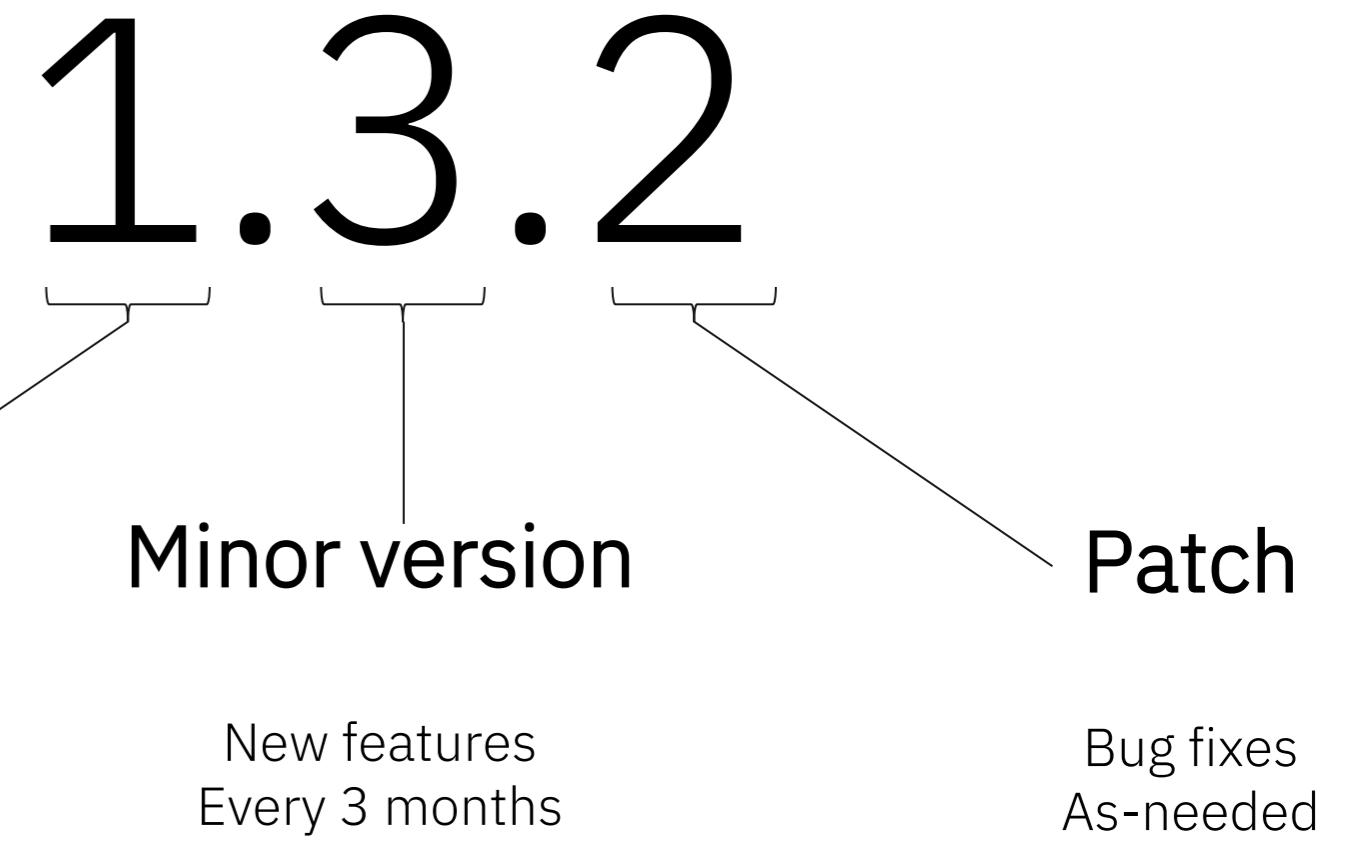
Ease of
use



Stability

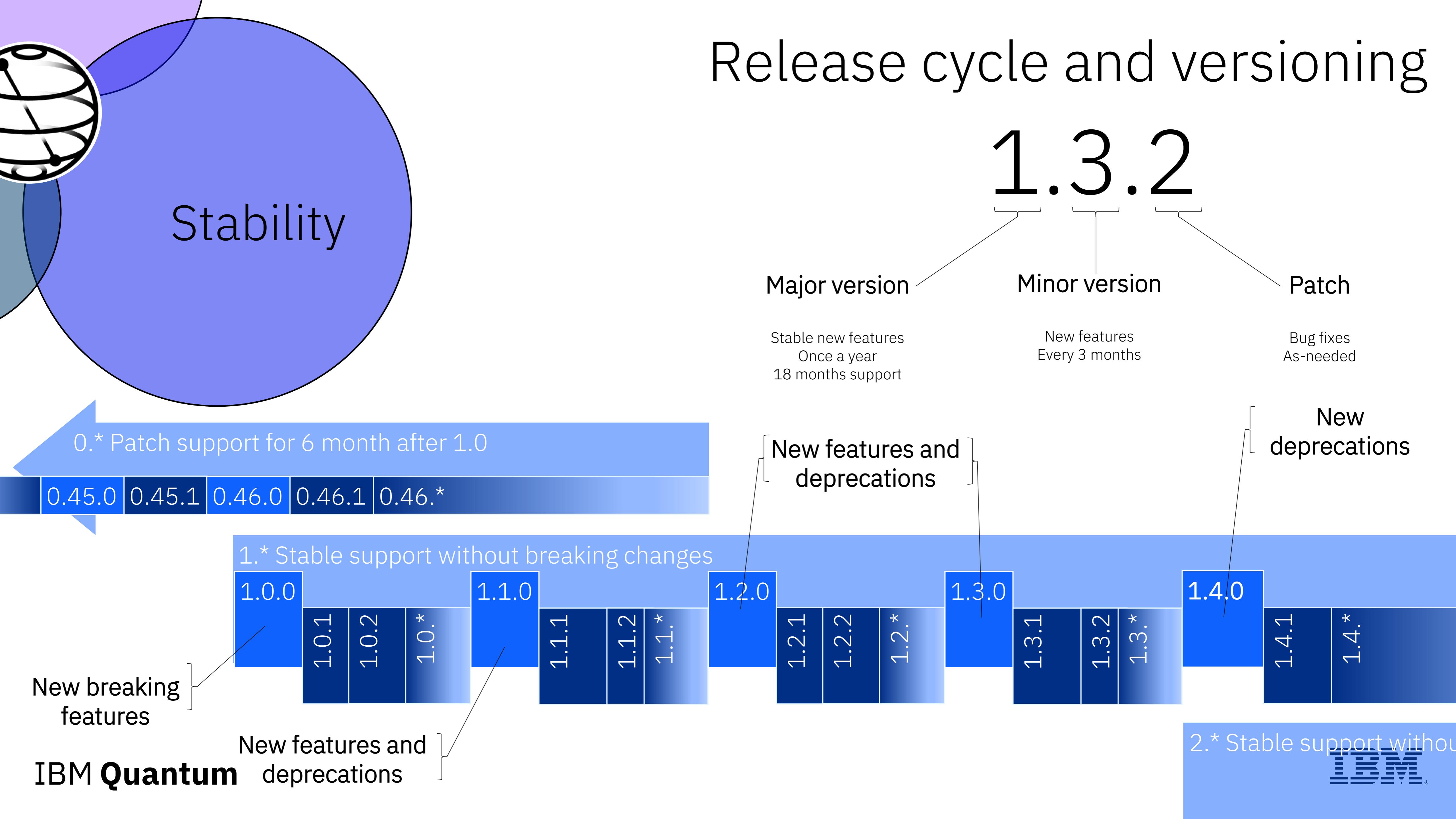


Release cycle and versioning

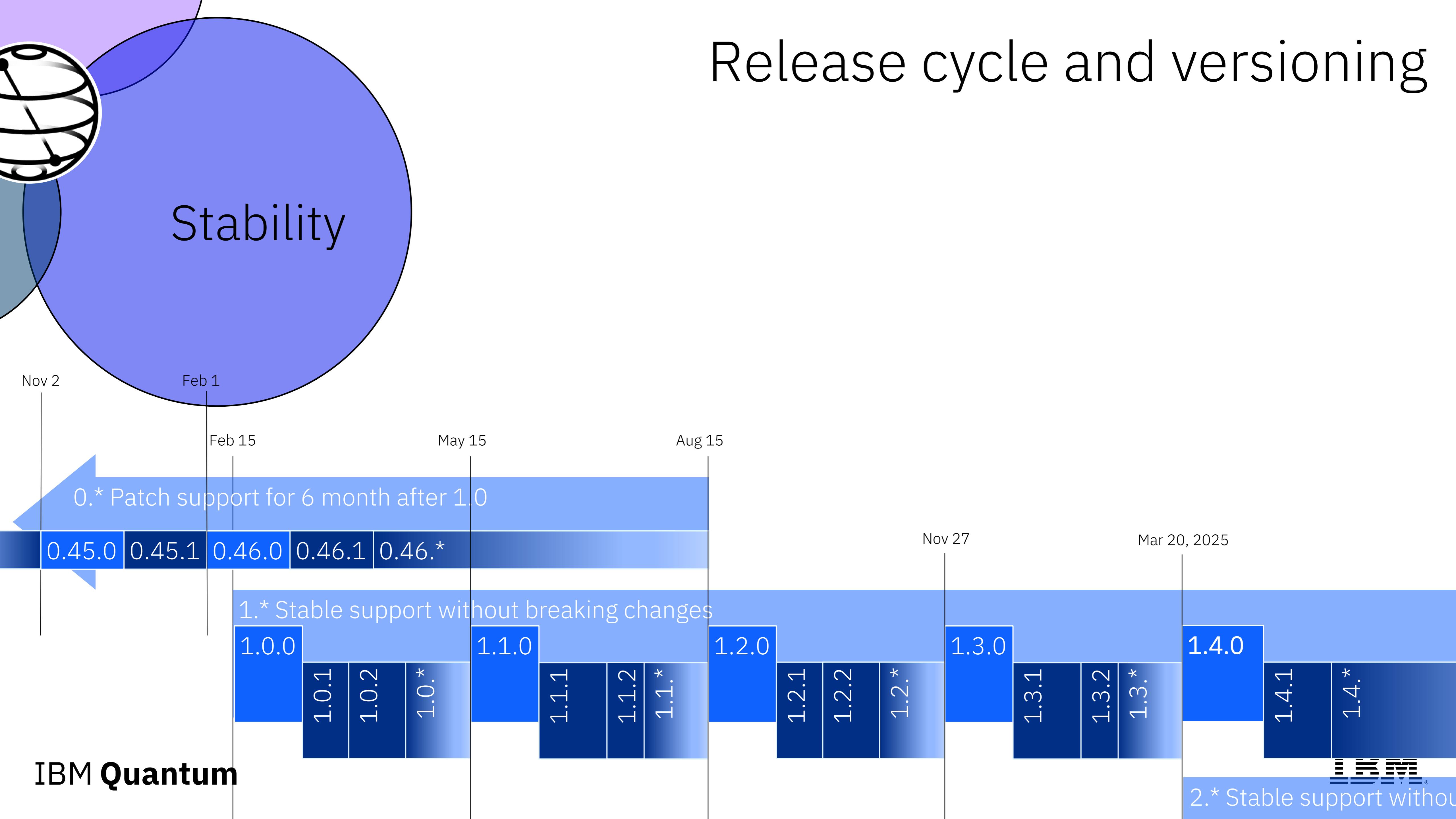


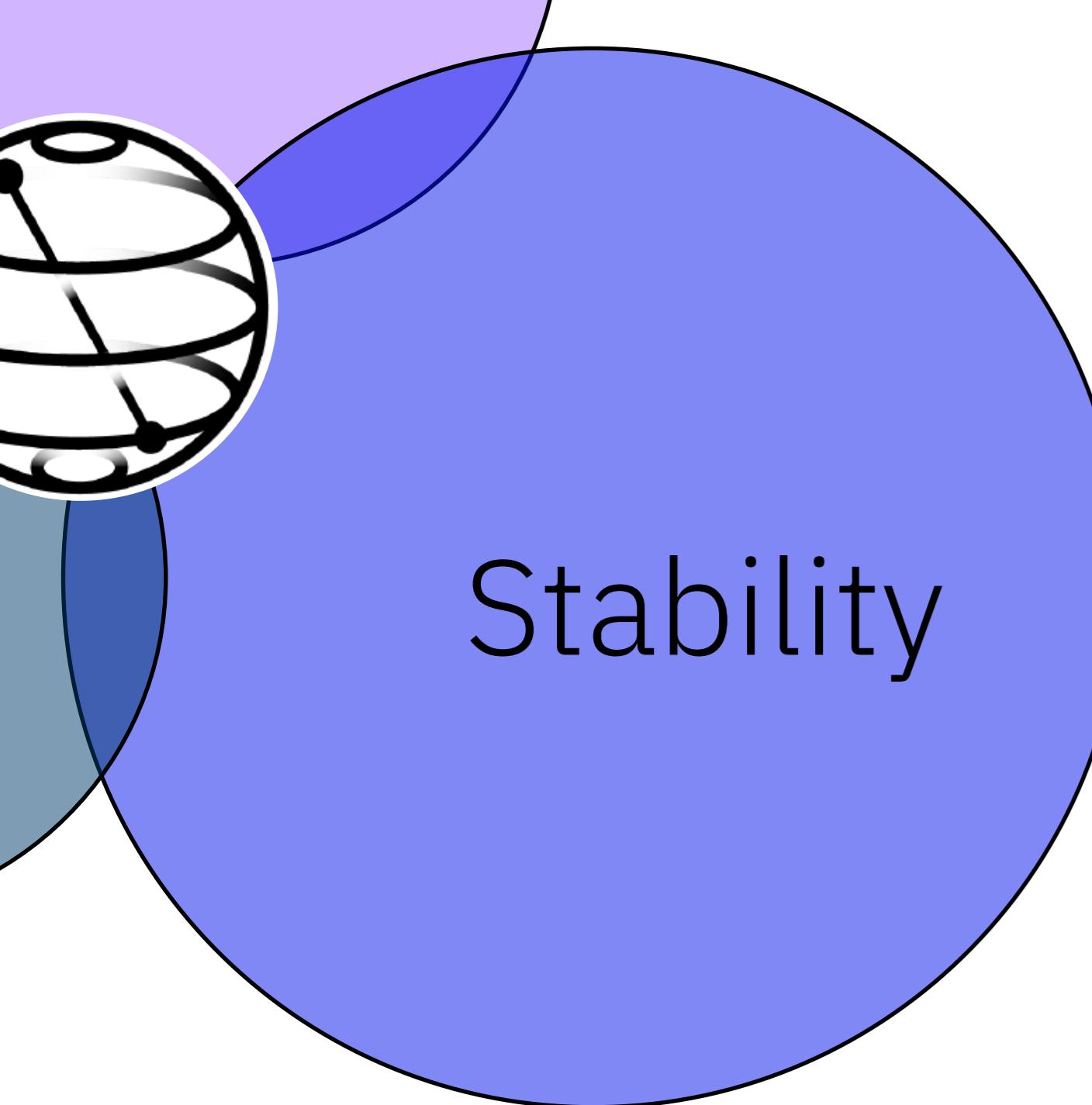
Semantic Versioning

Release cycle and versioning

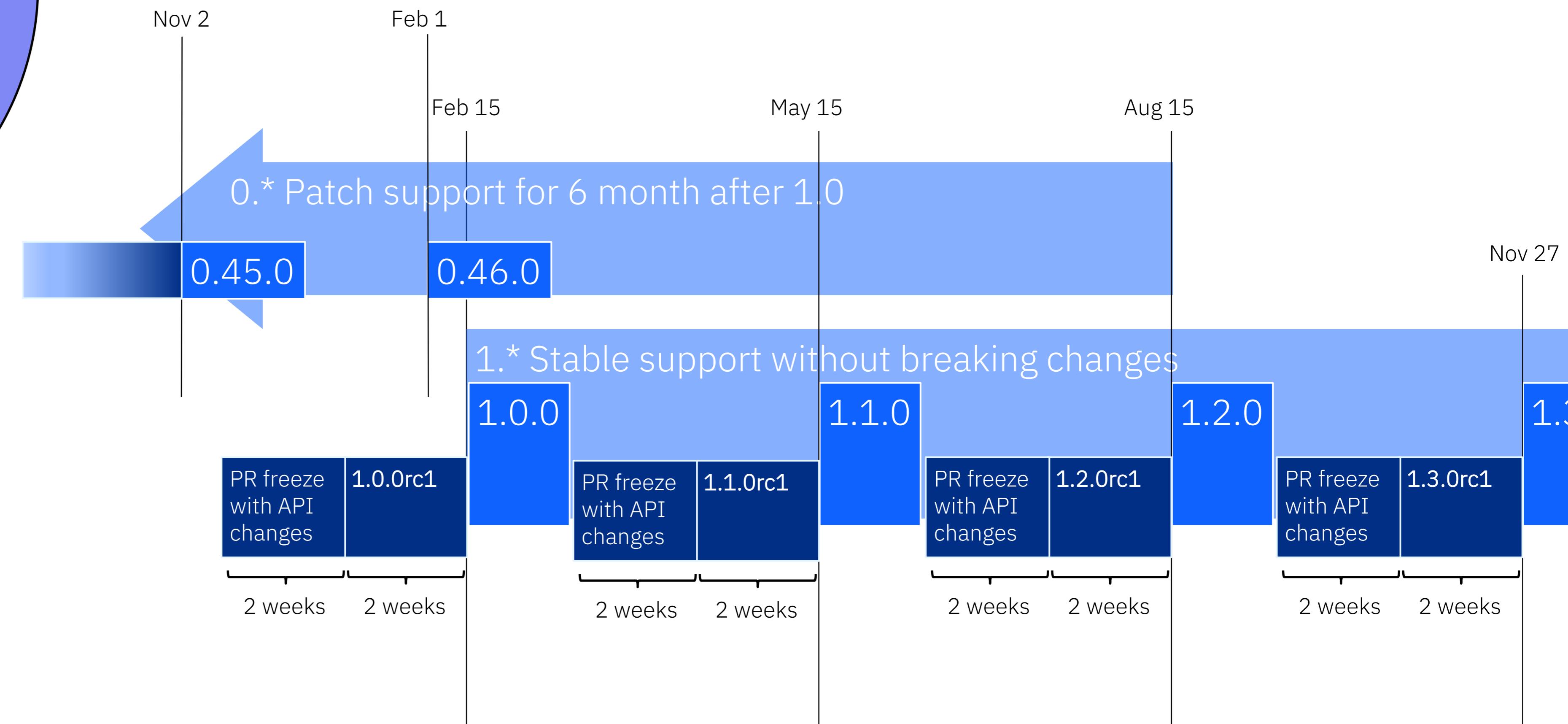


Release cycle and versioning

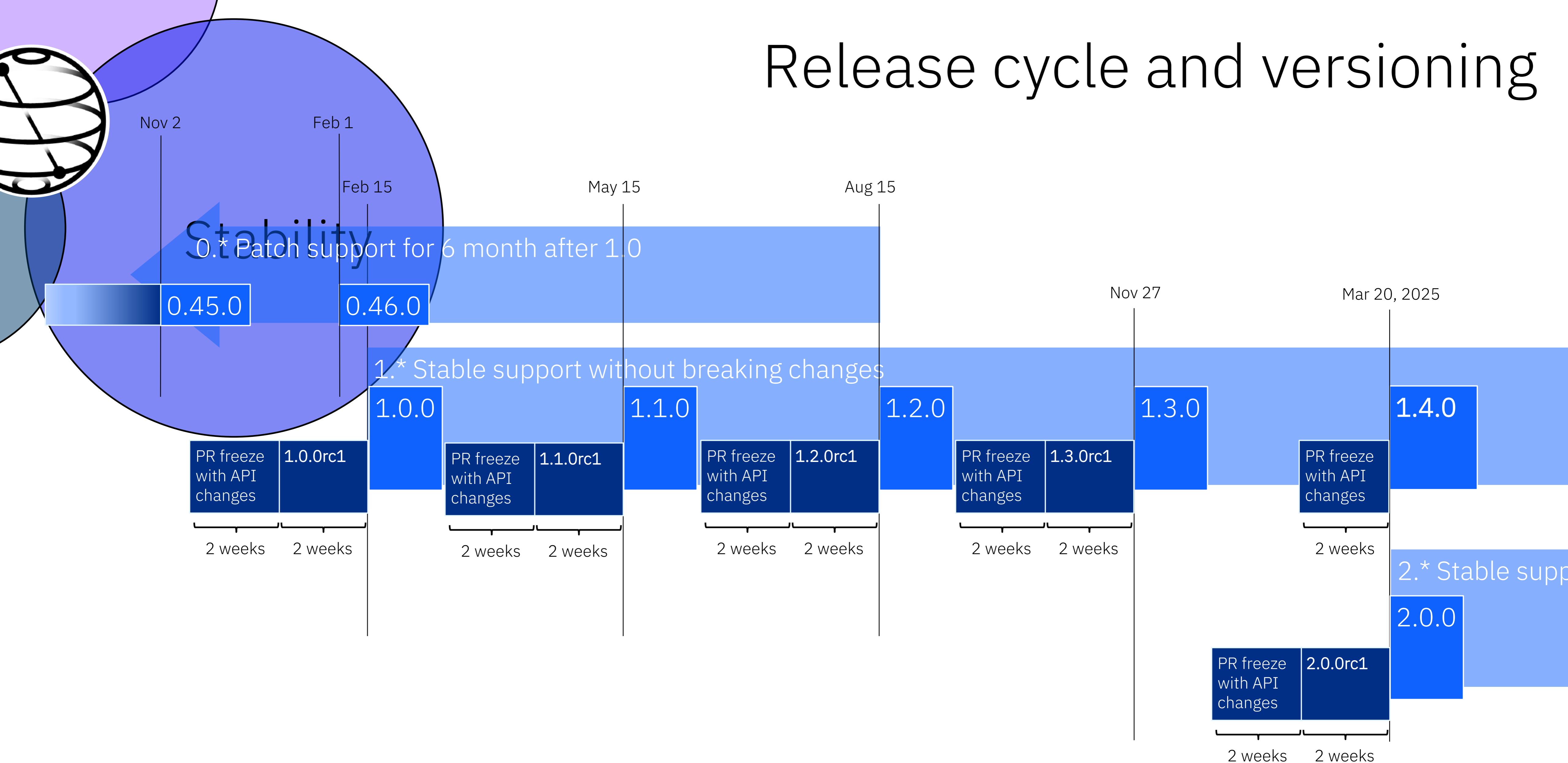


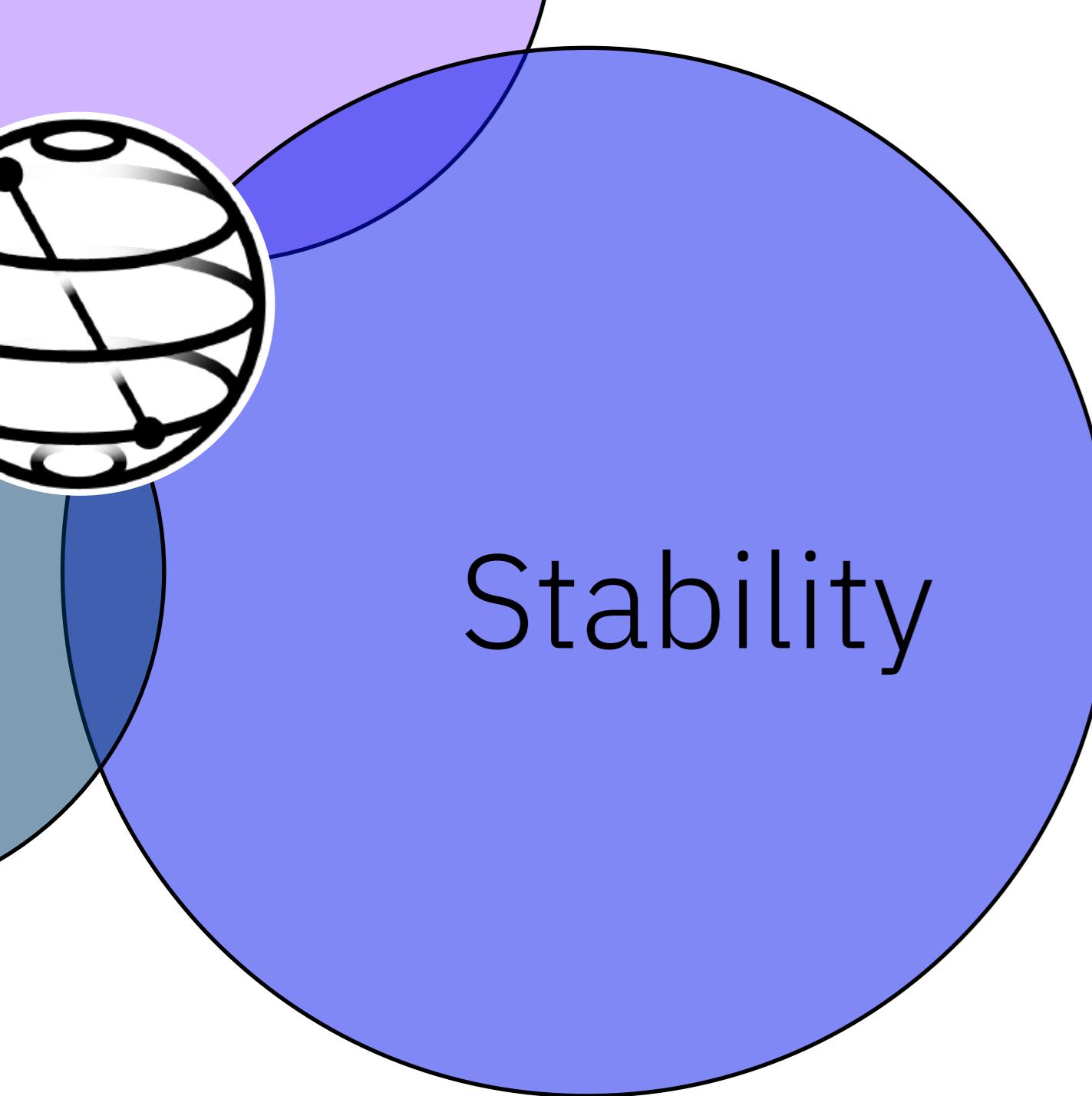


Release cycle and versioning



Release cycle and versioning



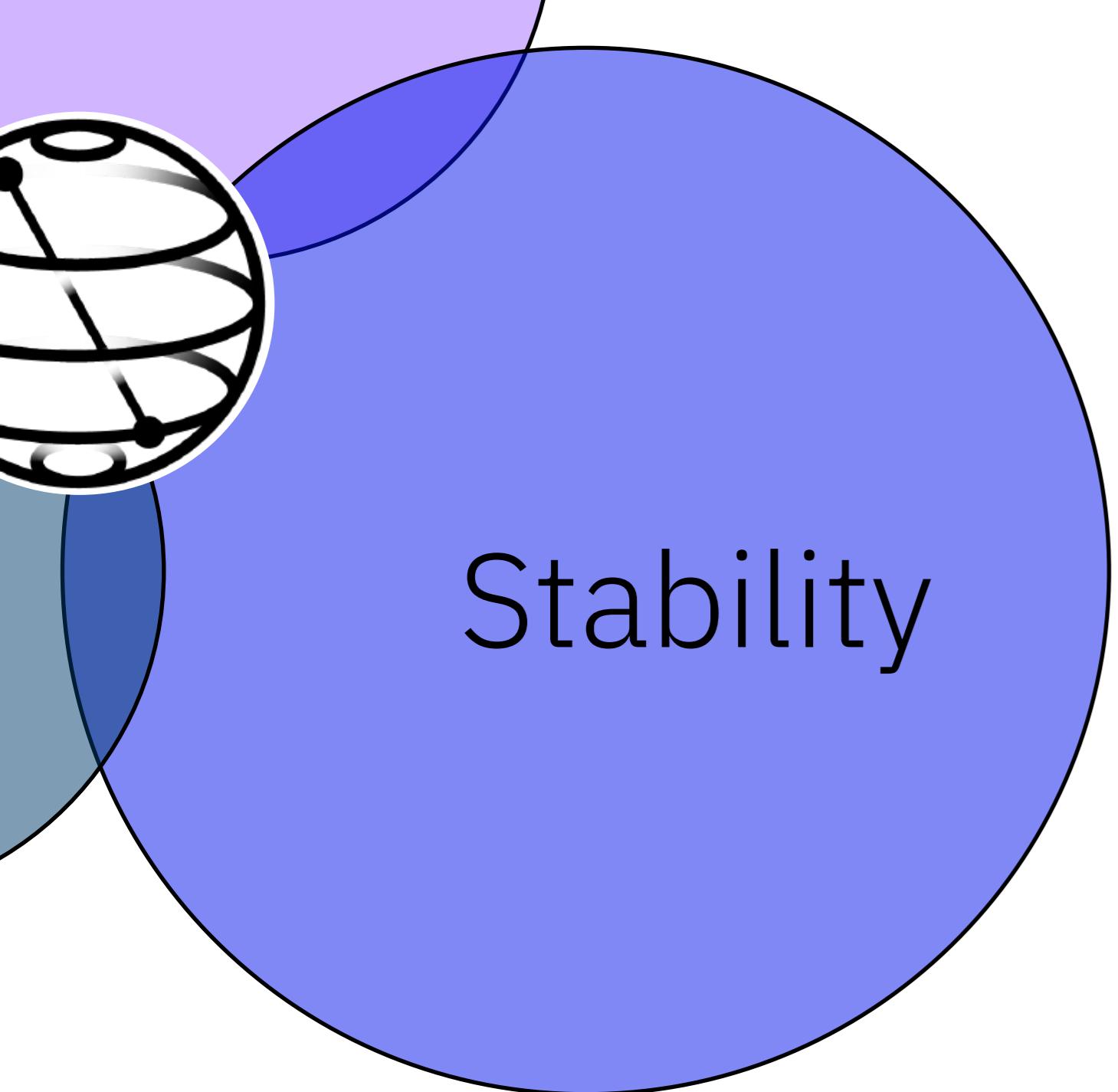


Check the GitHub milestones

<https://github.com/Qiskit/>

Release cycle and versioning

| Sort ▾ | |
|---|--|
| 7 Open | 43 Closed |
| 0.45.2 | No due date Last updated 2 days ago |
| 0.46.0 | Due by February 01, 2024 Last updated about 22 hours ago |
| Feature, Deprecation, and Removal Proposal Deadline: January 18, 2024 | |
| 1.0.0 | Due by February 15, 2024 Last updated 8 minutes ago |
| Feature, Deprecation, and Removal Proposal Deadline: January 18, 2024 | |
| Target RC1 date: Feb 1, 2024 | |
| 1.1.0 | Due by May 15, 2024 Last updated 2 days ago |
| Feature, Deprecation, and Removal Proposal Deadline: April 17, 2024 | |
| 95% complete | 1 open 23 closed |
| 37% complete | 25 open 15 closed |
| 40% complete | 90 open 60 closed |
| 0% complete | 2 open 0 closed |



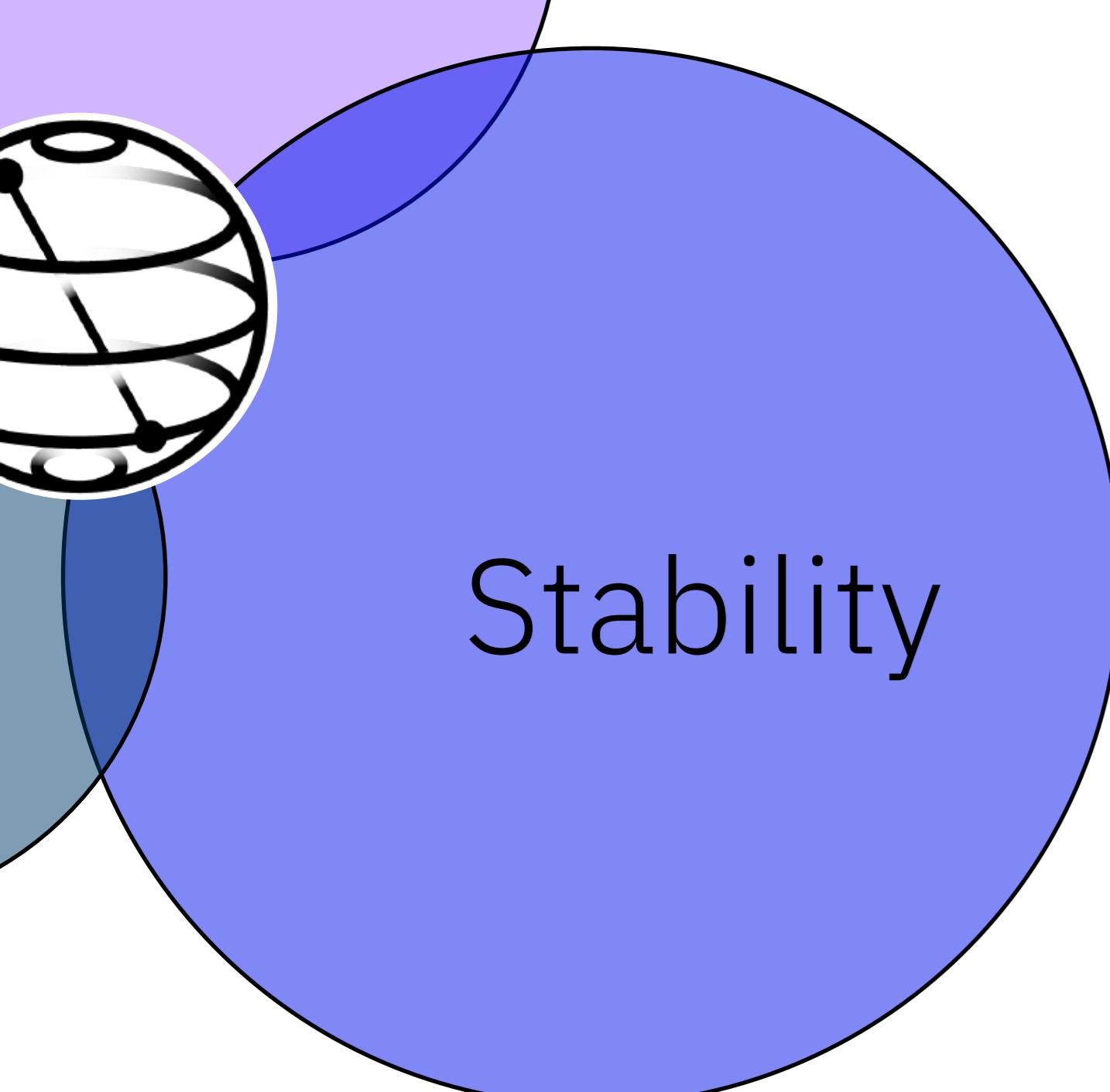
Release cycle and versioning

Stability

.. or subscribe to this
calendar

<webcal://airtable.com/app8A6ME5Gynn2TWf/shrwlHQ9rN3ts9AmN/iCal>



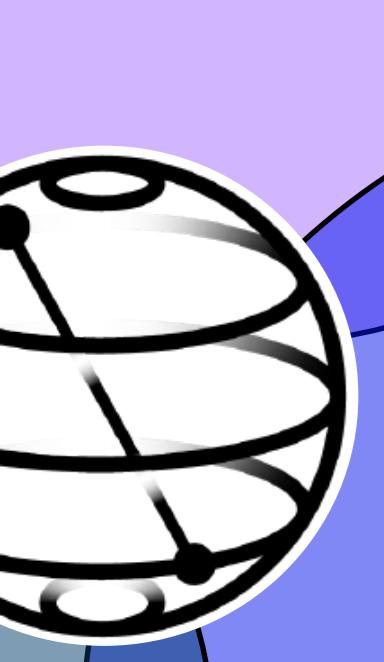


Migration guides

<https://docs.quantum.ibm.com/migration-guides/qiskit-1.0>

Removals in 1.0

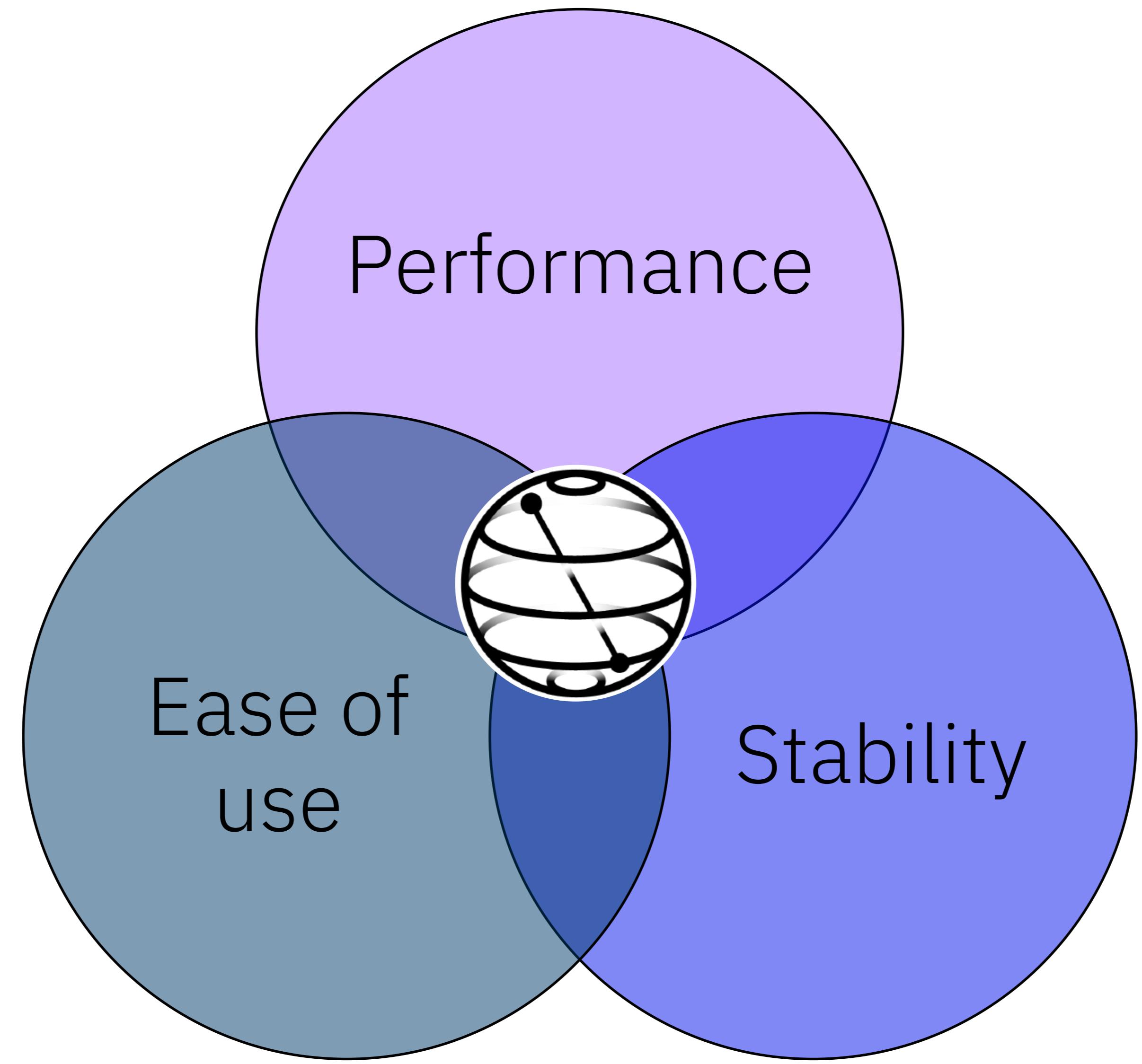
- ✓ Remove `execute()` function
- ✓ Remove `qiskit.algorithms` module
- ✓ Remove `qiskit.extensions` module
- ✓ Remove `qiskit.tools` module
- ✓ Remove `qiskit.test` module
- ✓ Move provider specific `FakeProviders` to the specific provider
- ✓ Rename `BasicAer` to `BasicSimulator`
- ✓ Remove legacy OpenQASM 2 parser
- ✓ Remove the discrete pulse library
- ✓ Remove practically all deprecated features



Stability

Qiskit Code Assistant: Migration

In the upcoming months we plan to build agents and new features to allow you to migrate your Qiskit code between versions, detect wrong code and provide potential fixes, or help you to work with the Qiskit functions.



Thank you for using Qiskit!

IBM Quantum

